UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Contribuições ao ensino de teste de software com o modelo flipped classroom e um agente conversacional

Leo Natan Paschoal

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Leo Natan Paschoal

Contribuições ao ensino de teste de software com o modelo flipped classroom e um agente conversacional

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Simone do Rocio Senger de Souza

USP – São Carlos
Maio de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

P279c

Paschoal, Leo Natan
Contribuições ao ensino de teste de software com o modelo flipped classroom e um agente conversacional / Leo Natan Paschoal; orientadora Simone do Rocio Senger de Souza. -- São Carlos, 2019.
300 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

1. Agente Conversacional. 2. Ensino de Computação. 3. Flipped Classroom. 4. Teste de Software. I. Souza, Simone do Rocio Senger de , orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Leo Natan Paschoal

Contributions to software testing education with the flipped classroom model and a conversational agent

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Simone do Rocio Senger de Souza

USP – São Carlos
May 2019

Este trabalho é dedicado à minha família, amigos, aos meus professores e à todos aqueles que acreditam que a educação é o melhor caminho.

AGRADECIMENTOS

Na minha concepção, existem diversos caminhos para se percorrer e conquistar o título de mestre. Alguns são mais acessíveis e outros são mais árduos. Particularmente, gosto de desafios e optei por uma formação mais sólida. Por ter me entregado completamente ao meio acadêmico, precisei de muita ajuda. Assim, nesta e na próxima lauda agradeço as pessoas que dedicaram um tempo de suas vidas para me ajudar a conquistar o título de mestre.

Ao final desta etapa de minha vida tenho que agradecer muitas pessoas, mas primeiramente, devo e agradeço a entidade que alguns chamam de Deus. Sem o Senhor, sem Sua luz, proteção e bênçãos, eu não conseguiria finalizar esse ciclo que tanto me esforcei para completar. Muito obrigado pela saúde, força de vontade e inspirações!

Aos meus pais Luiz e Neiva por terem compreendido que eu não tenho afinidades para trabalhar com agricultura e/ou pecuária, aceitando que eu fizesse um curso de graduação bem diferente daquilo que eles haviam planejado e posteriormente permitirem que eu me mudasse de estado para dar continuidade aos meus estudos, realizando o mestrado. Saibam que sem o apoio de vocês, eu nunca teria chegado aqui. Vocês formaram os degraus para que eu pudesse caminhar. Peço desculpas por ter sido responsável pelas diversas abdicações que tiveram que fazer ao longo de suas vidas.

Aos meus avós Altino e Elma por compreenderem as minhas ambições, sempre se preocuparem se eu estava conseguindo dormir, comer, estava bem de saúde e se eu precisava de alguma coisa. Eu sou muito grato a vocês. Desculpe-me por não estar presente nos momentos que vocês mais precisavam e por não ter acompanhado os momentos de descanso de vocês.

Aos meus irmãos Léia e Luan e ao meu cunhado Uiliam, por terem acreditado no meu potencial, ajudando-me nos momentos em que precisava de auxílio e por suprirem minhas ausências com a família. Por terem sido a principal fonte de informação para saber como o pai e a mãe estavam e por terem cuidado deles. O meu muito obrigado se estende às minhas sobrinhas Antonella e Luísa, por trazerem mais alegria à nossa família e ânimo para esse titio coruja durante a escrita deste extenso trabalho.

A minha amiga, eterna professora e orientadora Patricia Mariotto Mozzaquatro Chicon por ter me apresentado à Ciência. Por ter me dado todo o suporte que necessitei para conseguir ingressar em um programa de mestrado e sobreviver. Se hoje estou

finalizando esse ciclo, certamente foi porque tive uma preparação inicial com ela.

A minha querida orientadora, professora Drª. Simone do Rocio Senger de Souza, a qual carinhosamente eu tomei a liberdade de chamar de "mamis" desde quando a conheci. Eu nem tenho palavras para agradecê-la. Foram tantas oportunidades que ela me deu. Foram tantas noites e finais de semana que eu a fiz trabalhar! Ao final desse ciclo, eu quero dizer que eu sou muito grato a ela por ter me oferecido todas as possibilidades para que eu pudesse realizar o mestrado, de uma maneira muito especial e única. Serei eternamente grato por me orientar de uma maneira particular, adaptada a esse jovem aspirante a pesquisador. Muito obrigado por ter se desafiado a trabalhar com Educação em Computação, pelas inúmeras correções nos artigos, relatórios, na qualificação, nos exercícios e atividades práticas, nos protocolos dos experimentos e mapeamentos sistemáticos e nessa gigantesca dissertação. Também sou muito grato por todos os puxões de orelha que foram sutis, mas sempre necessários. A sua dedicação em ensinar é um exemplo que eu pretendo seguir. Certamente, aprendi e espero continuar aprendendo muito mais com a senhora. Se hoje estou finalizando um ciclo que comecei há dois anos, o mérito não é unicamente meu, mas sem sombra de dúvidas dessa pessoa, amiga e exemplar orientadora que sempre foi a mamis <3. Muito obrigado por tudo! É um privilégio poder trabalhar contigo!

Aos integrantes do Laboratório de Engenharia de Software (LabES) e Laboratório de Sistemas Distribuídos e Programação Concorrente (LaSDPC), em especial meus amigos: Ana Paula, Ana Spengler, Armando, Brauner, Bruno, Camila, Daniel, Danillo, Damasceno, Diógenes, Faimison, Francielle, Henrique, Ítalo, João Choma, João Paulo, Kamila, Lilian, Lina, Lívia, Maria Lydia, Misael, Pedro, Raul, Rodolfo, Sidgley, Silvana, Stevão e Vinicius Aires pelos momentos de descontração e companhia durante essa jornada.

Aos alunos de iniciação científica, por terem apoiado o desenvolvimento de alguns mecanismos de apoio que estão descritos no decorrer desta dissertação: Lucas Turci, Samuel Gamito.

Aos meus amigos do Rio Grande do Sul, que mesmo de longe me acompanharam nessa trajetória e contribuíram com a minha formação, em especial: Aliane, Cíndia, Fabrício, Felipe, Jéssica, Leonam, Mariane, Michele, Myke e Suelen.

Aos professores do Instituto de Ciências Matemáticas e da Computação (ICMC) da Universidade de São Paulo (USP), em especial, Drª. Elisa Yumi Nakagawa, Drª. Ellen Francine Barbosa, Dr. Marcio Delamaro, Drª. Rosana Teresinha Braga, Dr. Paulo Cesar Masiero, Dr. Paulo Sérgio de Souza e Dr. Seiji Isotani.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Código de Financiamento 001, e posteriormente ao suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), sob o processo nº 2017/10941-8.

♫ *"Let go and just be free"* ♫

RESUMO

PASCHOAL, L. N. **Contribuições ao ensino de teste de software com o modelo flipped classroom e um agente conversacional**. 2019. 300 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Teste de software é uma atividade do processo de desenvolvimento de software que busca revelar os defeitos presentes no software. Apesar dos constantes avanços e surgimento de diferentes modelos pedagógicos, este assunto continua sendo ensinado com o modelo tradicional de ensino. A comunidade de educação em computação vem investigando a aplicação de diferentes modelos pedagógicos e, nessa perspectiva, emerge o modelo pedagógico *flipped classroom* como uma interessante solução. Estudos mencionam sobre a necessidade de investigá-lo, buscando reconhecer seu potencial no ensino de um determinado conteúdo. Além disso, implementar o modelo exige uma intensificação de esforços para o professor, pois ele precisará planejar atividades e mecanismos de apoio ao ensino para cada um dos momentos previstos pelo modelo. Trabalhos relatam que os alunos durante o momento de estudo "antes da aula" podem ter dificuldade em entender aspectos teóricos sobre o conteúdo, pois esse pode ter conceitos que podem gerar dupla interpretação. Este trabalho de mestrado tem o propósito de contribuir com o ensino de teste de software, investigando o impacto do modelo *flipped classroom* no ensino de teste de software. O projeto também propõe uma abordagem de apoio ao planejamento e implantação desse modelo e analisa o uso de um agente conversacional como mecanismo de apoio ao *flipped classroom*. Estudos experimentais foram realizados visando entender o impacto do modelo no processo de aprendizado de teste de software. Na sequência, foram analisados conteúdos de teste de software ensinados em cursos de computação no Brasil, definidos objetivos de aprendizagem para esses conteúdos, selecionados mecanismos de apoio e atividades para contemplar os três momentos previstos pelo modelo e implementado um mecanismo para apoiar a implantação do modelo *flipped classroom* no ensino de teste. Por fim, um agente conversacional foi proposto e investigado no contexto de uma aula de teste de software em que o modelo *flipped classroom* foi utilizado. Os resultados deste trabalho indicam que o modelo *flipped classroom* permite que os alunos tenham melhores resultados no aprendizado, especialmente quando há realização de atividades práticas. Os alunos revelaram que preferem este modelo em detrimento ao tradicional. No entanto, o modelo produz uma carga de trabalho maior para o aluno. Por fim, o uso do agente conversacional como um mecanismo de apoio durante o momento "antes da aula" indica sua relevância para o *flipped classroom*.

Palavras-chave: Agente Conversacional. Ensino de Computação. Flipped Classroom. Teste de Software.

ABSTRACT

PASCHOAL, L. N. **Contributions to software testing education with the flipped classroom model and a conversational agent**. 2019. 300 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Software testing is an activity of the development process that aims to reveal the faults present in the software. Despite of constant progress and emergence of different pedagogical models, this subject is still taught with strategies based mainly on the traditional model. Although software testing is considered a relevant subject, some students are not very motivated to learn it. The computer education community has investigated the use of different teaching models and, from this perspective, the Flipped Classroom pedagogical model has emerged as an interesting solution. Several studies mention that is required to investigate this model in order to identify its potential in teaching a specific subject and/or teaching area. In addition, the implementation of this model demands an intensification of efforts from the teacher because he/she needs to plan activities and support mechanisms for all phases of this model: "before the class", "during the class", and "after the class". Results from the literature indicate that students may present difficulties in understand the theory during the phase "before the class"because studying alone can lead to many doubts and misunderstandings. Therefore, this master's thesis investigates the impact of the Flipped Classroom model on teaching of testing techniques and criteria. Experimental studies were initially undertaken aiming to understand the model's impact in the learning process of software testing. Following, we analyzed what subjects of software testing are taught in Computer courses in Brazil. Based on the results, we defined learning objectives for these subjects, selected support mechanisms and activities to contemplate the three phases of the Flipped Classroom model. An approach, named FlipClass-STT, to support the implementation of this model was defined and experimentally evaluated. Finally, the conversational agent TOB-STT was proposed and evaluated in context of a software testing class, in which the Flipped Classroom model was employed. The results of this work indicate that Flipped Classroom allows the students to have better learning outcomes, especially when performing practical activities. The students have revealed that they prefer this model against the traditional one. However, it is also observed that model produces a greater workload for the student. The use of TOB-STT how a support mechanism during the phase "before the class" indicate its relevance and impact for learning software testing.

Keywords: Computer Science Education. Conversational Agent. Flipped Classroom. Software Testing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Conjunto de atividades planejadas e realizadas 32
Figura 2 – Cenário típico da atividade de teste 39
Figura 3 – Exemplo de um *template* de planejamento de *Flipped Classroom* 49
Figura 4 – Exemplo de interação entre Cocorote e aluno 53
Figura 5 – ELAI interagindo com o aluno 54
Figura 6 – Interface principal do agente Ubibot 55
Figura 7 – Materiais didáticos disponíveis no ambiente Moodle 67
Figura 8 – Exemplo de um vídeo utilizado para instruir os estudantes 68
Figura 9 – *Box plot* dos escores das categorias do IMMS 77
Figura 10 – Comparação entre os escores de motivação dos grupos 78
Figura 11 – Grau de motivação dos sujeitos, considerando o grupo controle (esquerda) e o grupo experimental (direita) 79
Figura 12 – *Box plot* que compara o desempenho dos alunos 84
Figura 13 – Comparação de conhecimento prévio versus desempenho no grupo controle e no grupo experimental 85
Figura 14 – Vídeos selecionados para a instrução invertida 98
Figura 15 – Box plot que compara o número de acertos dos alunos nos testes de conhecimento 108
Figura 16 – Relação do tempo gasto pelos estudantes com o tempo médio gasto pelo grupo experimental 110
Figura 17 – Comparação entre o tempo gasto pelos alunos do grupo experimental com o tempo médio gasto pelos alunos do grupo controle 110
Figura 18 – Respostas dos estudantes sobre suas percepções relacionadas ao modelo *Flipped Classroom* 111
Figura 19 – Avaliação sobre a qualidade geral de itens do modelo *Flipped Classroom* 113
Figura 20 – *Box plot* para eficiência, eficácia e efetividade 132
Figura 21 – Processo adotado na condução do *survey* 143
Figura 22 – Taxonomia de Bloom e Flipped Classroom 152
Figura 23 – Planejamento da implementação de Flipped Classroom para Teste Funcional 165
Figura 24 – Planejamento da implementação de Flipped Classroom para Critérios baseados em Fluxo de Controle 165

Figura 25 – Planejamento da implementação de Flipped Classroom para Critérios baseados em Fluxo de Dados 166
Figura 26 – Planejamento da implementação de Flipped Classroom para Teste de Mutação 166
Figura 27 – Constituintes das atividades que integram a abordagem FlipClass-STT 167
Figura 28 – Sequência de ações para implementação da abordagem. Versão maximizada disponível em: <http://bit.ly/2QzeY3g> 168
Figura 29 – Modelagem do escopo do sistema Catalog-STT 171
Figura 30 – Diagrama de Casos de Uso do sistema Catalog-STT 173
Figura 31 – Estrutura de metadados adotada 175
Figura 32 – Interface da página inicial do sistema Catalog-STT 177
Figura 33 – Interface de registro de REA do sistema Catalog-STT 177
Figura 34 – Interface de consulta de REA do sistema Catalog-STT 178
Figura 35 – Grau de conhecimento dos participantes em Engenharia de Software . . 189
Figura 36 – Grau de conhecimento dos participantes em Teste de Software 190
Figura 37 – Experiência dos estudantes em utilizar os critérios de Teste Funcional em projetos de software 190
Figura 38 – Experiência dos estudantes em utilizar os critérios de Teste Estrutural em projetos de software 191
Figura 39 – Experiência dos estudantes em utilizar o critério de Teste de Mutação em projetos de software 191
Figura 40 – Comparação entre a preferência por materiais didáticos digitais e materiais didáticos impressos 192
Figura 41 – Tendência por recursos educacionais 193
Figura 42 – Frequência de acertos nos testes de conhecimento sobre Critérios baseados em Fluxo de Controle 195
Figura 43 – Frequência de acertos nos testes de conhecimento sobre Critérios baseados em Fluxo de Dados 196
Figura 44 – Frequência de acertos nos testes de conhecimento sobre teste baseado em defeitos – Teste de Mutação 198
Figura 45 – Frequência de acertos nos testes de conhecimento sobre Teste Funcional 200
Figura 46 – Frequência de alunos que desistiram da disciplina 204
Figura 47 – Frequência de alunos que foram aprovados/reprovados na disciplina . . 205
Figura 48 – Frequência dos alunos nas aulas 206
Figura 49 – Média final dos alunos 207
Figura 50 – Estruturação do conhecimento do agente TOB-STT 213
Figura 51 – Principais elementos que constituem o agente TOB-STT 214
Figura 52 – Interface do TOB-STT 216
Figura 53 – Exemplos de conversa nas três fases de uma conversação 224

Figura 54 – Estrutura que representa uma fração do conhecimento do TOB-STT . 228
Figura 55 – Histograma do tempo de duração das interações 229
Figura 56 – Histograma do número de perguntas feitas pelos usuários 230
Figura 57 – Proporção de cada tipo de resposta identificada 232
Figura 58 – Histograma dos anos de experiência dos participantes 233

LISTA DE TABELAS

Tabela 1 – Estatística descritiva da pontuação dos participantes no questionário IMMS . . . 76
Tabela 2 – Teste de Mann-Whitney – Categoria de Atenção . . . 80
Tabela 3 – Teste de Mann-Whitney – Categoria de Relevância . . . 80
Tabela 4 – Teste de Mann-Whitney – Categoria de Confiança . . . 81
Tabela 5 – Teste t de Student – Categoria de Satisfação . . . 81
Tabela 6 – Teste de Mann-Whitney - Motivação . . . 82
Tabela 7 – Estatística descritiva do conhecimento prévio dos participantes . . . 83
Tabela 8 – Estatística descritiva do desempenho dos alunos . . . 83
Tabela 9 – Teste de Mann-Whitney - Desempenho acadêmico dos alunos . . . 86
Tabela 10 – Resumo das respostas às questões de pesquisa relacionadas a variável motivação . . . 88
Tabela 11 – Exemplo de questões que constituem os testes de conhecimento . . . 101
Tabela 12 – Número de acertos e medidas de aprendizado para os modelos pedagógicos . . . 106
Tabela 13 – Estatística descritiva do tempo gasto pelos alunos do grupo experimental 108
Tabela 14 – As nove primeiras questões que constituem o formulário de percepção . 111
Tabela 15 – Resumo das respostas às questões de pesquisa do segundo experimento 116
Tabela 16 – Estatística descritiva dos valores de eficiência, eficácia e efetividade dos participantes . . . 131
Tabela 17 – Teste t de Student - eficiência . . . 133
Tabela 18 – Teste de Mann-Whitney - eficácia . . . 134
Tabela 19 – Teste de Mann-Whitney - efetividade . . . 134
Tabela 20 – Resumo das respostas às questões relacionada a efetividade . . . 136
Tabela 21 – Objetivos do *survey* . . . 143
Tabela 22 – Níveis de teste de software ensinados . . . 148
Tabela 23 – Técnicas de teste de software ensinadas . . . 148
Tabela 24 – Critérios de teste ensinados, agrupados por técnica . . . 149
Tabela 25 – Critérios de teste ensinados, agrupados por técnica . . . 150
Tabela 26 – Os objetivos de aprendizagem da abordagem FlipClass-STT . . . 154
Tabela 27 – Descrição sobre os cursos MOOC sobre teste de software . . . 158
Tabela 28 – Análise dos vídeos instrucionais sobre teste de software . . . 159
Tabela 29 – Análise dos recursos educacionais abertos sobre teste de software . . . 161

Tabela 30 – Análise dos objetos de aprendizagem sobre teste de software 163
Tabela 31 – Descrição dos metadados 176
Tabela 32 – Descrição dos metadados (Continuação) 176
Tabela 33 – Resumo do desenvolvimento da disciplina 181
Tabela 34 – Questões que compõem o formulário de caracterização de *expertise* dos estudantes 182
Tabela 35 – Detalhes sobre as disciplinas analisadas 203
Tabela 36 – Quantidade de alunos não desistentes 205
Tabela 37 – Classificação dos conteúdos conversados entre TOB-STT e os alunos . 231
Tabela 38 – Questionário produzido para coleta das percepções dos pesquisadores . 233
Tabela 39 – Respostas dos pesquisadores nas cinco perguntas 234
Tabela 40 – Síntese das observações realizadas pelos pesquisadores 235
Tabela 41 – Síntese das sugestões de melhorias apontadas pelos pesquisadores . . . 236
Tabela 42 – Questionário para coletar as percepções dos estudantes sobre o agente conversacional como mecanismo de apoio ao momento “antes da aula” 238
Tabela 43 – Percepções dos estudantes sobre o agente conversacional durante o momento “antes da aula” 238

SUMÁRIO

1 INTRODUÇÃO 25
1.1 Contexto e Motivação 25
1.2 Problemática da Investigação 30
1.3 Objetivos 31
1.4 Procedimentos Metodológicos 31
1.5 Organização do Trabalho 35

2 FUNDAMENTAÇÃO TEÓRICA 37
2.1 Considerações Iniciais 37
2.2 Teste de Software 37
2.2.1 Técnicas e Critérios de Teste 40
2.2.2 Ensino de Teste de Software 44
2.3 Flipped Classroom 46
2.3.1 Implementação de Flipped Classroom 48
2.3.2 Vantagens e desvantagens de Flipped Classroom 49
2.4 Agentes Conversacionais 51
2.4.1 Agentes no Ensino de Computação 52
2.4.2 Tecnologias para o Desenvolvimento de Agentes 56
2.5 Considerações Finais 59

3 ESTUDOS EXPERIMENTAIS 61
3.1 Considerações Iniciais 61
3.2 Primeiro Experimento 62
3.2.1 Definição do Experimento 62
3.2.2 Planejamento 64
3.2.3 Operação 69
3.2.4 Análise dos Resultados 72
3.2.5 Síntese das Análises 86
3.2.6 Ameaças à Validade 89
3.2.7 Apresentação e Empacotamento 92
3.3 Segundo Experimento 92
3.3.1 Definição 93
3.3.2 Planejamento 94

3.3.3 Operação . . . 99
3.3.4 Análise dos Resultados . . . 104
3.3.5 Síntese das Análises . . . 115
3.3.6 Ameaças à Validade . . . 116
3.3.7 Apresentação e Empacotamento . . . 120
3.4 Terceiro Experimento . . . 120
3.4.1 Definição . . . 121
3.4.2 Planejamento . . . 122
3.4.3 Operação . . . 125
3.4.4 Análise dos Resultados . . . 128
3.4.5 Síntese das Análises . . . 135
3.4.6 Ameaças à Validade . . . 136
3.4.7 Apresentação e Empacotamento . . . 137
3.5 Considerações Finais . . . 138

4 FLIPCLASS-STT . . . 139
4.1 Considerações Iniciais . . . 139
4.2 Estabelecendo a abordagem FlipClass-STT . . . 141
4.2.1 Definição de conteúdo . . . 141
4.2.2 Definição dos objetivos de aprendizagem . . . 151
4.2.3 Definição de recursos educacionais . . . 154
4.2.4 Definição de atividades . . . 162
4.3 FlipClass-STT: uma visão geral . . . 163
4.4 Publicação e Aplicação da abordagem FlipClass-STT . . . 169
4.5 Aprimoramento da FlipClass-STT . . . 169
4.5.1 Catalog-STT: um Catálogo de Recursos Educacionais Abertos sobre Teste de Software . . . 170
4.6 Considerações Finais . . . 178

5 APLICAÇÃO DA ABORDAGEM FLIPCLASS-STT . . . 179
5.1 Considerações Iniciais . . . 179
5.2 Proposta Didática . . . 180
5.3 Coleta dos Dados . . . 181
5.4 Relato de Experiência . . . 184
5.5 Resultados . . . 187
5.5.1 Expertise dos estudantes . . . 189
5.5.2 Formato de Recursos Educacionais . . . 192
5.5.3 Aprendizado . . . 193
5.6 Lições Aprendidas . . . 200
5.7 Apresentação e Empacotamento . . . 202

5.8 Paralelo Histórico 202
5.9 Considerações Finais 207

6 O AGENTE CONVERSACIONAL TOB-STT 209
6.1 Considerações Iniciais 209
6.2 Agentes Conversacionais para Educação 210
6.3 Visão Geral do Agente Conversacional 211
6.3.1 Delimitação do Conhecimento 211
6.3.2 Características e Implementação 213
6.4 Avaliação do TOB-STT 227
6.5 TOB-STT: uma experiência de uso 237
6.6 Considerações Finais 241

7 CONCLUSÕES 243
7.1 Considerações Iniciais 243
7.2 Principais descobertas 243
7.3 Contribuições 244
7.4 Limitações 245
7.5 Trabalhos Futuros 246
7.6 Publicações 247

REFERÊNCIAS 251

APÊNDICE A MAPEAMENTO SISTEMÁTICO SOBRE AGENTES CONVERSACIONAIS PARA EDUCAÇÃO 269

CAPÍTULO

1

INTRODUÇÃO

1.1 Contexto e Motivação

Em decorrência de falhas históricas envolvendo sistemas de software que não conseguem satisfazer seus usuários, a qualidade do software se tornou um atributo chave para um produto de software (LAND; HIGGS, 2007). A falha de um software, por sua vez, normalmente está relacionada a problemas no processo de desenvolvimento. O processo de desenvolvimento de software não pode ser considerado uma tarefa trivial, pois envolve desde a tradução das necessidades do usuário para um documento de requisitos, a transformação dos requisitos elicitados em um projeto de software, a codificação do projeto em uma linguagem que seja possível ser interpretada ou compilada por um computador, o teste do código implementado, e a entrega do software para uso operacional (SINGH; GAUTAM, 2016). Assim, por envolver atividades humanas, este processo não está livre de conter falhas.

O teste de software é uma atividade essencial no desenvolvimento de software, sendo considerada uma atividade de engenharia de software que é responsável por oferecer subsídios à garantia da qualidade do software. Teste de software tem o propósito de revelar a presença de defeitos no software (MYERS; SANDLER; BADGETT, 2011). Os defeitos presentes no software afetam sua qualidade, podendo reduzir a confiança do usuário (SHIPPEY, 2015) e gerar custos econômicos. Um estudo publicado pela *University of Cambridge* afirma que os defeitos presentes em software custam à economia dos Estados Unidos $ 312 bilhões por ano (BRADY, 2013). Diante disso, as organizações se submetem a executar atividades de teste de software de forma sistemática, uma vez que se o software não for testado sua qualidade pode ficar comprometida (CLEGG; ROJAS; FRASER, 2017).

Conduzir o teste de software de maneira sistemática exige um conhecimento ade-

quado dos profissionais responsáveis pela realização da atividade de teste. No entanto, pesquisas indicam que muitos egressos de cursos superiores de computação são contratados pela indústria de software com conhecimentos excepcionais em codificação de software, porém com pouca habilidade em teste de software (ASTIGARRAGA *et al.*, 2010). É comum que egressos de cursos de formação superior em computação se formem e sejam inseridos no mercado de desenvolvimento de software com poucas habilidades em testar programas (CLARKE *et al.*, 2014), uma vez que o teste de software é pouco abordado durante o ensino de processo de desenvolvimento de software (LEMOS *et al.*, 2015). A indústria reconhece que, muitas vezes, os profissionais que trabalham com engenharia de software, contratados diretamente da Universidade, não tiveram treinamento formal adequado em teste de software (CHAN *et al.*, 2005).

Apesar dos diversos vestígios sobre a carência de profissionais qualificados na área de teste de software, relacionados com a formação de estudantes de cursos de educação superior em computação, outros fatores estão associados, como a não trivialidade da tarefa de testar (DELAMARO; MALDONADO; JINO, 2016) e a desvalorização dos profissionais de teste (VALLE, 2017). Por esses motivos, a utilização sistemática do teste de software ainda não é uma prática comum, principalmente em instituições brasileiras (BARBOSA *et al.*, 2007). Todavia, a indústria de software vem percebendo que o teste de software é uma atividade indispensável dentro do processo de desenvolvimento de software. Assim, o meio acadêmico tem a difícil missão de atender a demanda existente, oferecendo meios para a formação e a capacitação pessoal, que abrangem aspectos teóricos e práticos de teste de software (BARBOSA *et al.*, 2007).

Buscando amenizar os problemas mencionados e estimular a disseminação do conhecimento na área, estudos vêm sendo realizados no contexto de educação em teste de software. Dentre os trabalhos existentes, há estudos que se concentram nos aspectos relativos à abordagem de ensino utilizada (PASCHOAL; SILVA; SOUZA, 2017). De acordo com Soska, Mottok e Wolff (2016), o meio acadêmico precisa ajustar os métodos de ensino para atender de maneira mais satisfatória à demanda da indústria de software, uma vez que a indústria deseja profissionais com habilidades técnicas, e que os mesmos possuam uma compreensão mais aprofundada sobre o assunto. Outras pesquisas também discutem sobre a ineficiência em ensinar os conteúdos de teste de software exclusivamente por abordagens tradicionais (SOUZA *et al.*, 2013). Segundo Smith *et al.* (2012), as abordagens educacionais utilizadas atualmente no ensino de teste de software são pouco eficientes, não sendo capazes de oferecer condições propícias para que os estudantes obtenham resultados significativos nos processos de aprendizagem.

Para contribuir com a formação do estudante de computação, Soska, Mottok e Wolff (2016) discorrem sobre investir em abordagens que promovam a aprendizagem ativa, isto é, que ofereçam condições para que o estudante de teste de software desenvolva auto-

nomia, pensamento crítico e criativo. Com abordagens de aprendizagem ativa, o estudante acaba por se envolver no processo de ensino e aprendizagem, por meio de atividades que direcionam o estudante para o protagonismo no seu processo de aquisição de conhecimento, proporcionando oportunidades para o sucesso profissional (CARVALHO; RAMOS, 2015). Tais atividades podem ser de leitura, escrita, discussão e resolução de problemas (SOSKA; MOTTOK; WOLFF, 2016). Nesse sentido, quando se consegue fazer com que o estudante construa novos conhecimentos, o processo de ensino e aprendizagem realmente é efetivo. No entanto, além de oferecer condições propícias para uma aprendizagem efetiva, por meio de modelos da aprendizagem ativa, é necessário promover um maior engajamento dos alunos.

Dentre os modelos pedagógicos que apoiam os processos de ensino e aprendizagem existentes, um que pode ser explorado e investigado no contexto de ensino de teste de software é o *flipped Classroom* (PASCHOAL; SILVA; SOUZA, 2017; PASCHOAL; SOUZA, 2018a). Esse modelo transfere a prioridade de aprendizagem do professor para o aluno. Neste sentido, ele fundamenta-se pela mudança das atividades passivas, normalmente realizadas em sala de aula, para fora da sala de aula (KIAT; KWONG, 2014). Para tanto, a exposição do conteúdo realizado tipicamente em sala de aula é substituída por instruções baseadas em computador que são realizadas fora da sala de aula, e o tempo em sala de aula é utilizado para discussão de conceitos complexos e dúbios, atividades práticas, resolução de problemas e atividades colaborativas (KERR, 2015).

No modelo *flipped classroom*, o aluno estuda preliminarmente o conteúdo em casa e em sala de aula ele aprofunda o seu aprendizado com atividades práticas orientadas pelo professor, o que remete em um deslocamento da atenção da aprendizagem do professor para o estudante, aumentando o envolvimento do estudante (DAVIES; DEAN; BALL, 2013). Desse modo, nessa transferência de prioridades, o estudante tem a possibilidade de estudar a *priori* o conteúdo de uma aula, por intermédio de materiais didáticos digitais oferecidos pelo professor. O professor, em sala de aula, tem o papel de supervisor, podendo direcionar a aprendizagem do estudante de uma maneira personalizada, completando e aprofundando o conhecimento, como também conduzindo o estudante na realização de atividades práticas em sala de aula. Assim, *flipped classroom* parte da premissa filosófica de estimular os estudantes a serem responsáveis por realizarem seus estudos em ambientes externos à sala de aula, preparando-se para executar atividades práticas em sala de aula.

Apesar de existirem alguns estudos sobre o uso do modelo *flipped classroom* em disciplinas de cursos de computação (GIANNAKOS; KROGSTIE; CHRISOCHOIDES, 2014) (*e.g.*, Introdução à Ciência da Computação (FASSBINDER *et al.*, 2015), Arquitetura de Computadores (GEHRINGER; PEDDYCORD III, 2013), dentre outros), o assunto têm sido pouco explorado no contexto de teste de software (PASCHOAL; SILVA; SOUZA, 2017). Nesse campo, acredita-se que *flipped classroom* poderia oferecer subsídios para os

estudantes compreenderem os métodos e os procedimentos que são adotados na área, porque é um modelo que visa a aprendizagem ativa. No entanto, antes de adotar o modelo é necessário testar se ele realmente é útil para a formação de estudantes de teste de software, avaliando seu impacto nos processos inerentes ao ensino e a aprendizagem. Uma forma de avaliar é por meio de experimentos (WOHLIN *et al.*, 2012).

A experimentação é uma prática utilizada em diversas áreas do conhecimento humano (KITCHENHAM *et al.*, 2002). No que se refere à *flipped classroom*, alguns estudos experimentais têm sido conduzidos a fim de comprovar a validade do modelo em determinados domínios (KOSTARIS *et al.*, 2017; AşıKSOY; ÖZDAMLı, 2016; TAO; HUANG; TSAI, 2016). No entanto, por ser um assunto contemporâneo, que vem sendo explorado em pesquisas recentes, em muitas áreas não existe conhecimento sobre o real efeito do uso do modelo. Existem muitos estudos a serem realizados com o modelo pedagógico *flipped classroom*, que vão além da conjunção das especificidades da área em que o modelo está sendo aplicado, com as características do modelo, uma vez que o mesmo acaba afetando os sujeitos envolvidos nos processos de ensino e aprendizagem.

Aliado à análise de viabilidade, é importante reforçar que a adoção de um novo modelo pedagógico em um curso ou disciplina, implicará ao professor uma maior dedicação para implantação. Assim, é possível descrever que o modelo exige esforços intensificados por parte do professor. O professor, por exemplo, precisa definir o que os alunos irão fazer nos três momentos que constituem o modelo: antes da aula, durante a aula, depois da aula (BERGMANN; SAMS, 2012). Além disso, precisa preparar materiais didáticos de instrução – porque, em geral, em *flipped classroom* os alunos recebem instruções baseadas em computador, atividades que serão realizadas pelos alunos em sala de aula, testes para averiguar o conhecimento obtido por parte dos alunos, dentre outros.

Com a inversão da sala de aula, os estudantes necessitarão de mais assistência, porque estarão expostos a cometerem equívocos e conflitos nos momentos de estudo (GANNOD; BURGE; HELMICK, 2008). Em aplicações de *flipped classroom*, os alunos frequentemente se deparam com dúvidas ao fazer a leitura do material didático ou ao assistir os vídeos recomendados pelo professor (RAMíREZ; HINOJOSA; RODRÍGUEZ, 2014), porque alguns conteúdos podem ter duplo sentido, ou até mesmo ser difícil de ser entendido. No que se refere ao tempo de dedicação para compreensão do conteúdo, acredita-se que as vídeos aulas são os recursos com maiores condições de apoiar, uma vez que cada aluno pode gastar o tempo que precisa para aprender, podendo assistir aos vídeos quantas vezes forem necessárias (BERGMANN; SAMS, 2012). Referente à assistência para tratar controvérsias e equívocos, as pesquisas salientam que as dúvidas devem ser solucionadas pelo professor em sala de aula (GANNOD; BURGE; HELMICK, 2008; FASSBINDER *et al.*, 2014; PASCHOAL; SILVA; SOUZA, 2017). No entanto, solucionar as dúvidas dos alunos somente durante a aula pode provocar um fenômeno não previsto no curso, em que os

alunos não estudam o conteúdo, por não ter apoio durante o aprendizado (PASCHOAL; SOUZA, 2018a). Assim, torna-se relevante investigar meios para apoiar o aluno durante o aprendizado. Dentre os mecanismos existentes, que possibilitam solucionar as dúvidas dos alunos antes da aula, destacam-se os agentes conversacionais.

Um agente conversacional pode ser definido como um programa de computador desenvolvido para simular uma conversa inteligente com um ou mais usuários (DERYUGINA, 2010), utilizando métodos auditivos ou textuais (SHAIKH *et al.*, 2015). Esse tipo de programa interage com seres humanos por intermédio de linguagem natural (GHOSE; BARUA, 2013), através de interfaces[1] (ABDELKEFI; KALLEL, 2016). O seu funcionamento consiste basicamente em interpretar entradas recebidas em linguagem natural e encontrar a saída mais adequada como solução de entrada (SHAIKH *et al.*, 2015). Nesse sentido, os agentes conversacionais processam as entradas dos usuários, com o intuito de reproduzir a resposta correta (DERYUGINA, 2010).

No contexto educacional, existe um grande interesse em agentes conversacionais (MORAES; SANTOS; REDECKER, 2016), uma vez que eles podem trazer benefícios aos estudantes. Dentre as vantagens, podem ser mencionadas a capacidade em atuar como um tutor que orienta o estudante durante a realização de atividades (MASSARO *et al.*, 2005; PASCHOAL *et al.*, 2016; PASCHOAL; CHICON; FALKEMBACH, 2017) de um curso, de uma disciplina ou de uma unidade da disciplina; apoiar a resolução de dúvidas que surgem quando o aluno realiza atividades (SILVA; COUTO; ODAKURA, 2012); auxiliar o aluno a aprender um determinado conceito (PRIMO *et al.*, 1998), instruindo-o sobre conteúdos específicos (BADA; MENEZES, 2012; HERPICH *et al.*, 2016); indicar materiais educacionais digitais em diferentes tipos de formato (LEONHARDT *et al.*, 2003; LEONHARDT; NEISSE; TAROUCO, 2003; RIBEIRO *et al.*, 2010); dentre outras.

Em virtude desses fatores, os agentes conversacionais podem ser instanciados para o domínio de teste de software. Pesquisas recentes como as de Rosa *et al.* (2017) vêm discorrendo sobre o uso dos mesmos em ambientes educacionais de aprendizagem, como mundos virtuais. No entanto, Rosa *et al.* (2017) desenvolvem o agente conversacional para apoiar o usuário durante a interação sobre conceitos associados a teste de software, não se aprofundando em alguns tópicos de teste de software, como ferramentas, fases, técnicas e seus respectivos critérios de teste de software. Além disso, o trabalho não segue uma filosofia de dados abertos (ISOTANI; BITTENCOURT, 2015), em que as bases de conhecimento estariam disponibilizadas publicamente, dificultando o reaproveitamento do agente conversacional.

[1] É o meio de comunicação pelo qual o usuário interage com o programa.

1.2 Problemática da Investigação

Diante do que foi apresentado anteriormente, nota-se a necessidade em experimentar novos modelos pedagógicos no ensino de teste de software, em especial, aqueles que buscam promover maiores condições para o aluno desenvolver autonomia, pensamento crítico e ao mesmo tempo que o aluno consiga ter uma boa formação teórico-prática, alinhando o estado da arte com as tecnologias emergentes da área. Diante disso, emerge um modelo denominado *flipped classroom*, o qual vem sendo adotado por organizações e instituições de ensino, como uma proposta para melhorar os índices de aproveitamento dos alunos, promover um maior engajamento nos alunos, estimular o aluno a ter mais interesse pela área, dentre outros.

Apesar de muitos estudos relatarem a adoção do modelo em diferentes áreas, outros estudos vêm descrevendo sobre a necessidade de investigar se o modelo realmente pode ser utilizado para ensinar um determinado assunto. Bergmann e Sams (2012) em seu estudo, reforçam a ideia de não implementar o modelo diretamente em um curso ou em uma disciplina, mas analisar a viabilidade do modelo, enquanto abordagem de ensino. Diante disso, foi formulada a primeira questão que esta dissertação se propôs a investigar: *É viável ensinar conteúdos de teste de software com o modelo pedagógico flipped classroom?*

Adotar o modelo pedagógico *flipped classroom* para ensinar um determinado conteúdo, conforme mencionado na seção anterior, exige uma intensificação de esforços por parte do professor, que precisará planejar todas as atividades e materiais didáticos necessários para cada um dos momentos previstos pelo modelo. Diante disso, não é suficiente apenas analisar se o modelo é viável para ensinar teste de software, mas em caso de sucesso do modelo, também é importante reconhecer e definir estratégias para apoiar o planejamento e implementação do modelo. Com base nessa perspectiva, é importante o estabelecimento de uma iniciativa para incentivar a sua implementação. No entanto, há um número expressivo de conteúdos para serem abordados, diferentes materiais para serem pensados, exercícios, atividades e trabalhos práticos para serem selecionados e/ou desenvolvidos. Assim, outra questão que permeia a presente dissertação surge: *Como implementar o modelo Flipped Classsroom no ensino de teste de software?*

Por fim, outro ponto que foi discutido na seção anterior foi o problema associado à preparação dos alunos durante o momento "antes da aula". Durante esse momento, estudos relatam que os alunos precisam de mecanismos para apoiá-los e há relatos que os materiais didáticos mais tradicionais como os vídeos, os slides e suas variações não consegue satisfazer de modo geral todos os alunos. Muitas vezes, os conteúdos geram dupla interpretações, possuem conceitos e definições não triviais para serem compreendidas. Esses fatores geram uma série de dúvidas que podem prejudicar o aprendizado dos alunos. Um mecanismo de apoio com potencial para minimizar as dúvidas dos alunos é o agente conversacional. Desse modo, a última questão que será abordada na presente dissertação

foi estruturada, a saber: *Um agente conversacional é útil para o modelo flipped classroom no contexto de uma aula de teste de software?*

1.3 Objetivos

Diante da problemática levantada, constituída por três questões, surgem os objetivos desta pesquisa. Esta dissertação de mestrado tem a finalidade de contribuir com a temática de ensino em teste de software. Para tanto, retrata uma investigação sobre a viabilidade do modelo *flipped classroom*, com o propósito de definir uma abordagem de apoio ao ensino de teste de software que se fundamenta nesse modelo pedagógico. Desse modo, a abordagem intitulada FlipClass-STT (um acrônimo para *An Approach to Software Testing Teaching with Flipped Classroom Model*) é estabelecida valorizando o estado da arte que apresenta uma discussão sobre alguns conteúdos serem mais adequados de ser ensinados com *flipped classroom* do que outros. Nessa perspectiva, a abordagem FlipClass-STT é desenvolvida a partir de estudos experimentais conduzidos com o propósito de analisar o impacto do modelo no ensino de teste de software. Essa abordagem busca orientar o professor sobre como conduzir os três momentos que constituem o modelo, indicar um conjunto de mecanismos de apoio e de atividades, com licença livre, que possam ser utilizadas pelos professores ao implementarem o modelo no ensino de teste. Adicionalmente, um agente conversacional será investigado como um mecanismo de apoio ao ensino de teste de software com *flipped classroom*.

1.4 Procedimentos Metodológicos

Considerando os objetivos estabelecidos, durante o desenvolvimento desta dissertação um conjunto de atividades foi pré-estabelecido e, posteriormente, desenvolvido. As atividades seguiram direcionamentos, diretrizes, processos e recomendações já estabelecidos na área, visando mitigar possíveis problemas que poderiam emergir durante o desenvolvimento do trabalho. Em síntese, um total de oito atividades foram realizadas, as quais são apresentadas na Figura 1 e descritas a seguir.

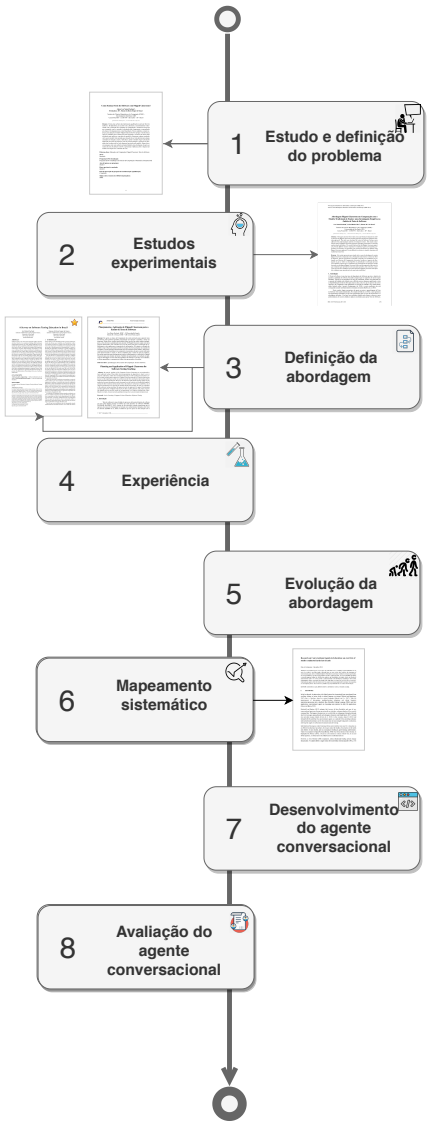


Figura 1 – Conjunto de atividades planejadas e realizadas

Estudo e definição do problema

A primeira etapa envolvida no estudo consistiu no entendimento da área e os problemas que permeiam a temática escolhida. Para compreensão do estado da arte que inclui as pesquisas sobre ensino em teste de software, uma revisão de literatura não sistemática foi realizada. Essa revisão retornou estudos sistemáticos conduzidos na área, em particular os estudos de Valle, Barbosa e Maldonado (2015), Valle, Barbosa e Maldonado (2015) e Lauvås e Arcuri (2018). Nesses estudos secundários, notou-se a discussão feita pelos autores sobre abordagens de ensino e a preocupação de pesquisadores e da comunidade em estabelecer meios para ajustar os métodos de ensino usados atualmente em aulas de teste de software. Diante disso, surgiu o modelo *flipped classroom* como uma proposta interessante, já investigado como modelo para ensinar conteúdos de outras disciplinas da computação.

Na sequência ao entendimento sobre a preocupação em contribuir e oferecer subsídios para o ensino e treinamento em teste de software, o modelo pedagógico *flipped classroom* foi estudado. Em especial, a partir da leitura do trabalho de Bergmann e Sams (2012), foi constatado que antes de adotar o modelo pedagógico é importante e relevante ter uma visão clara dos possíveis benefícios e limitações do modelo em relação ao ensino de determinado conteúdo. Diante disso, emergiu o primeiro problema que esta dissertação visa abordar, ou seja, compreender se *flipped classroom* é um modelo que fará com que o aluno consiga aprender.

Em conjunto com esse problema, Bergmann e Sams (2012) também mencionam sobre o processo de planejamento e implementação do modelo, descrevendo que o trabalho do professor se intensifica ao escolher utilizá-lo para ensinar um determinado conteúdo. No trabalho de Bishop e Verleger (2013), os autores discorrem sobre o esforço exigido por parte do professor para a efetiva implementação do modelo em um curso. Assim, o segundo problema que foi reconhecido e que será abordado nesta dissertação (conforme mencionado na seção 1.2) consiste em reconhecer os estágios necessários para planejar e implementar *flipped classroom* em disciplinas/cursos de teste de software.

Durante o entendimento sobre o modelo *flipped classroom* e o impacto de sua implementação, foi observado que alguns estudos revelam sobre a necessidade de intensificar o apoio ao aluno no momento em que ele estará estudando por conta própria, sem apoio do professor. A lacuna observada consiste na falta de apoio ao aluno quando ele precisa estudar fora da sala de aula. Esse problema, normalmente associado a definições, terminologias e conceitos, poderia ser solucionado por meio de tecnologias baseadas em conhecimentos, como os sistemas inteligentes. Diante disso, emerge a proposta de experimentar um agente conversacional como um mecanismo de apoio ao *flipped classroom*.

Estudos experimentais

Para responder a primeira questão de pesquisa explanada na Seção 1.2, estudos experimentais foram planejados e executados. Foram realizados três estudos experimentais, com grupos de alunos variados, matriculados em diferentes disciplinas de cursos de graduação em Computação do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP). Três estudos experimentais foram necessários, pois durante a realização deste trabalho de mestrado foi surgindo a necessidade de analisar diferentes aspectos, como desempenho do aluno, motivação, ganho de aprendizagem, preferência por abordagem usada no ensino e efetividade do aluno em realizar a atividade de teste a partir do aprendizado com o modelo. Para condução dos estudos experimentais, o processo estabelecido por Wohlin *et al.* (2012) foi seguido.

Definição da abordagem

A partir do reconhecimento das oportunidades oferecidas pelo modelo *flipped clas-*

sroom ao ensino de teste de software, visando auxiliar os professores da área a implementar o modelo em disciplinas de engenharia de software, teste de software ou disciplinas relacionadas, foi estudado e definido uma abordagem para apoiar o planejamento e implementação do modelo para esta área de ensino. Essa abordagem é uma iniciativa para reagir a segunda questão de pesquisa levantada neste trabalho. O processo de definição envolveu a identificação de assuntos que seriam contemplados pela abordagem, entendimento sobre o planejamento para cada um dos momentos previstos e atividades, definição de objetivos de aprendizagem para os conteúdos, seleção de materiais didáticos de apoio, dentre outras. Os direcionamentos de Bergmann e Sams (2012) para implementação do modelo foram seguidos buscando definir a abordagem.

Experiência

Após a definição preliminar da abordagem, ela foi experimentada, visando obter um entendimento melhor sobre os desafios de implementar a abordagem no ensino de teste e os benefícios da implementação, observados por meio da verificação do impacto no aprendizado dos alunos. A experiência englobou os conteúdos de teste que foram definidos na abordagem FlipClass-STT. Em adição ao relato de experiência, uma análise histórica foi realizada, comparando os resultados de disciplinas de graduação sobre teste de software que foram ministradas exclusivamente pelo modelo tradicional de ensino e os resultados decorrentes de uma disciplina de teste de software ministrada exclusivamente a partir da adoção da abordagem FlipClass-STT. Essa análise busca trazer novas perspectivas e lições sobre a adoção do modelo.

Evolução da abordagem

Com base na experiência adquirida e lições aprendidas com a implementação do modelo durante os estudos de caso e ao ministrar a disciplina ministrada exclusivamente com *flipped classroom*, necessidades de melhoria foram observadas, especialmente associadas à falta de flexibilidade para a implantação do modelo. Diante disso, foram definidas algumas mudanças na abordagem, visando evoluí-la. Especialmente é dado atenção a automatização da implementação da abordagem, por intermédio do estabelecimento de um mecanismo de apoio à seleção de materiais didáticos e atividades. Esse mecanismo, intitulado Catalog-STT, foi desenvolvido seguindo uma adaptação de metadados do padrão *IEEE Learning Object Metadata* e além de apoiar a seleção dos recursos busca manter sustentável a abordagem.

Mapeamento sistemático

Considerando a lacuna observada que emergiu na questão relacionada ao uso de um agente conversacional em uma aplicação de *flipped classroom*, foi realizado um estudo com o propósito de identificar quais os agentes conversacionais existentes na literatura que poderiam contribuir com o aprendizado de teste de software. Para tanto, um ma-

peamento sistemático foi planejado e conduzido. Nesse mapeamento foram investigadas diferentes questões de pesquisa, visando oferecer um panorama sobre o desenvolvimento de agentes conversacionais para a educação. O mapeamento foi conduzido seguindo as recomendações apresentadas nos estudos de Petersen *et al.* (2008), Petersen, Vakkalanka e Kuzniarz (2015). Os resultados dele oferecem subsídios para o desenvolvimento de um agente conversacional.

Desenvolvimento de um agente conversacional

Para contribuir com o ensino de teste de software e analisar a questão *"Um agente conversacional é útil para o modelo flipped classroom no contexto de uma aula de teste de software?"*, um agente conversacional foi implementado, utilizando as tecnologias reveladas por meio do mapeamento sistemático. O desenvolvimento do agente consistiu no estabelecimento de pares conversacionais e uma avaliação com especialistas da área de teste de software, em que foi considerado durante avaliação a capacidade do agente conversacional representar o conhecimento da área. Para apoiar a avaliação do agente, intitulado TOB-STT (*a chatBOT to support Software Testing Teaching*) foram adotados alguns critérios de avaliação para sistemas de diálogo em linguagem natural discutidos nos estudos de AbuShawar e Atwell (2016), Paschoal, Oliveira e Chicon (2018) e Herpich *et al.* (2016).

Avaliação do agente conversacional

Por fim, a última etapa da pesquisa foi avaliar o agente conversacional enquanto mecanismo de apoio ao *flipped classroom*. Para conduzir esta avaliação, foi estabelecido um estudo em uma disciplina em que um tópico de teste de software estava sendo ministrado. O agente conversacional foi oferecido como mecanismo de apoio adicional. Buscou-se reconhecer os benefícios/limitações do uso do agente conversacional em conjunto com o *Flipped Classrrom* por meio do *feedback* emitido por estudantes que interagiram com o agente conversacional quando dúvidas surgiam ao estudarem o conteúdo.

1.5 Organização do Trabalho

Neste capítulo, foram discorridos o contexto, as motivações e lacunas, os objetivos e as principais contribuições alcançadas com a execução deste trabalho de mestrado. No Capítulo 2, é apresentada uma revisão de literatura envolvendo os principais temas que são discutidos nesta dissertação, que são: teste de software, ensino de teste de software, *flipped Classroom* e agentes conversacionais. O Capítulo 3 apresenta os estudos experimentais conduzidos. O Capítulo 4 descreve a abordagem FlipClass-STT, incluindo a definição dos assuntos que são contemplados, a definição dos objetivos de aprendizagem, materiais didáticos e atividades que foram selecionados, bem como um mecanismo de apoio à seleção de recursos educacionais abertos sobre teste de software que visa apoiar a implementação

da abordagem. O Capítulo 5 apresenta um relato de experiência, assim como uma análise histórica sobre o uso do modelo *flipped Classroom* em uma disciplina de teste de software em comparação com o uso do modelo tradicional em um disciplina de teste de software. O Capítulo 6 apresenta a concepção, implementação e avaliação de um agente conversacional para apoiar o ensino de teste de software, incluindo uma análise sobre a utilidade de um agente conversacional como mecanismo de apoio ao *flipped Classroom*. Por fim, no Capítulo 7, são discutidas as conclusões, descritas as publicações realizadas ao longo do projeto, bem como indicações de estudos futuros.

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações Iniciais

Este capítulo aborda, de uma maneira geral, os principais temas desta pesquisa, são eles: o teste de software, o ensino de teste de software, o modelo *flipped classroom* e também os agentes conversacionais. Desse modo, o capítulo está organizado da seguinte forma. Na Seção 2.2 é discutida a atividade de teste, as terminologias envolvidas, o cenário típico da atividade, as principais técnicas e critérios usados durante a atividade, assim como uma síntese sobre contribuições feitas pela comunidade de computação para subsidiar o ensino teste de software. Na Seção 2.3 é apresentado o modelo *flipped classroom*, sua relação com o modelo tradicional de ensino, algumas discussões sobre contribuições existentes para o planejamento e implementação do modelo, bem como as vantagens e desvantagens associadas a sua implementação. Na sequência, a Seção 2.4 descreve os agentes conversacionais enquanto mecanismos de apoio ao aprendizado, em que são abordadas as terminologias associadas, seus propósitos quando usados na temática educacional, alguns agentes conversacionais usados em trabalhos que buscam contribuir com o ensino de computação e, também, um conjunto de tecnologias utilizadas no desenvolvimento desses sistemas de software. Por fim, na Seção 2.5 são apresentadas as conclusões do capítulo.

2.2 Teste de Software

Por muitos tempo, a qualidade do software dependia da construção cuidadosa de um projeto de software e das habilidades de implementação do programador (PRESSMAN; MAXIM, 2016). Nesse sentido, poucas estratégias eram adotadas para conter os defeitos presentes no software. No entanto, em decorrência do advento de áreas como a engenharia de software, que dentre seus objetivos visa à produção de software de alta qualidade; técnicas e atividades especializadas e eficientes foram emergindo, fornecendo subsídios

para o desenvolvimento de software com maior qualidade (MALDONADO, 1991). Dentre as atividades que fazem parte do processo de desenvolvimento de software, o teste de software busca revelar defeitos antes do software ser disponibilizado para uso.

Na atividade de teste de software é comum encontrar na literatura da área alguns termos, tais como: erro, defeito, falha, engano, dentre outros. Apesar de serem utilizados em outros domínios e em situações informais do dia a dia de Engenheiros de Software e programadores, como sinônimos, no âmbito da literatura tradicional de teste de software eles não são sinônimos e cada um desses termos possui um significado. Com o objetivo de definir claramente o real significado de cada termo, o IEEE (*Institute of Electrical and Electronics Engineers*) (THE INSTITUTE OF ELECTRONICAL AND ELECTRONICS ENGINEERS, 1990) os padronizou, criando o padrão IEEE 610.12. A definição de cada termo é descrita a seguir:

- **Defeito (em inglês: *fault*)**: passo, processo ou definição de dados incorretos (*i.e.* comando incorreto em um programa).
- **Engano (em inglês: *mistake*)**: ação humana que ocasiona em um resultado incorreto (*i.e.* ação incorreta feita pelo programador).
- **Erro (em inglês: *error*)**: diferença entre o valor obtido e o valor esperado (*i.e.* qualquer resultado não esperado na execução de um programa).
- **Falha (em inglês: *failure*)**: produção de uma saída incorreta com relação à especificação (*i.e.* um erro pode ocasionar uma falha).

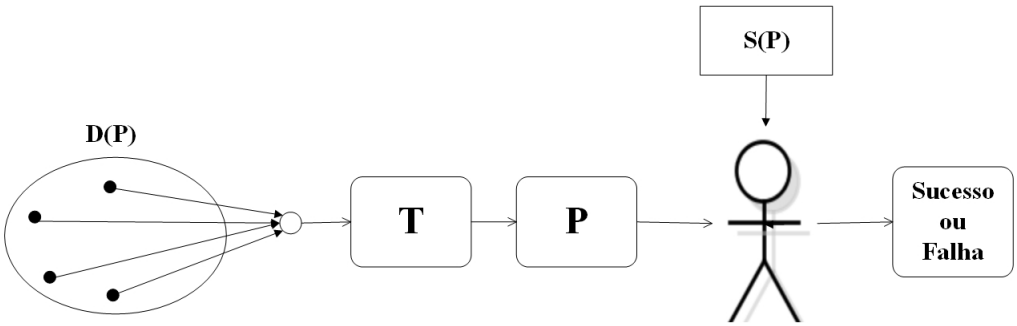
Mesmo com a existência do padrão IEEE 610.12, segundo Delamaro, Maldonado e Jino (2016), essas definições nem sempre são seguidas pelos pesquisadores e profissionais da área. Todavia, é importante apresentar esses conceitos e terminologias, a fim de deixar o leitor conscientizado que os mesmos existem. Hetzel (1987) discorre que outros termos também são comuns na área e merecem ser apresentados. Delamaro, Maldonado e Jino (2016) também reforçam a importância de apresentar a definição adequada de tais termos. Desse modo, a seguir serão expostas algumas terminologias que são consideradas comuns na área e que por consequência disso são utilizadas neste trabalho:

- **Domínio de entrada**: é representado pelo conjunto de valores que podem ser utilizados para executar um dado programa. O domínio de entrada envolve todos os valores possíveis para um determinado programa.
- **Domínio de saída**: é representado pelo conjunto de resultados produzidos por um dado programa. O domínio de saída envolve todos os resultados que foram produzidos por um determinado programa.

- **Dado de Teste**: são os dados de entrada e as condições associados a um determinado caso de teste.
- **Caso de Teste**: são elementos que definem os dados de entrada do teste, as condições de execução e as saídas esperadas a partir da execução da condição.
- **Conjunto de Teste**: representa o grupo de todos os casos de teste produzidos durante uma atividade de teste.
- **Oráculo**: É o instrumento que verifica se a saída obtida a partir da execução do programa ou software em teste coincide com a saída esperada pelo caso de teste. O oráculo pode ser uma pessoa ou uma ferramenta que automatiza a atividade.

A partir das definições apresentadas, é possível compreender o cenário típico de uma atividade de teste de software. Delamaro, Maldonado e Jino (2016) apresentam o seguinte cenário. Um programa *(P)* é composto por domínios de entrada *(D)* e domínios de saída *(S)*. O domínio de entrada é representado por um conjunto de valores que pode ser utilizado para executar o programa, denotado por *D(P)*. Já o domínio de saída é representado por um conjunto de todos os possíveis resultados produzidos pelo programa, denotado por *S(P)*. Assim, um dado de teste do programa *(P)* é um elemento do domínio de entrada. Um par formado por um dado de teste mais um elemento do domínio de saída, representa um caso de teste. Assim, quando se tem diversos pares de caso de teste, têm-se um conjunto de casos de teste *(T)*. Ao definir T, executa-se P em teste com T e verifica-se qual é o resultado obtido. Se o resultado obtido for igual às saídas esperadas, não foi possível identificar defeitos. Em contraposição, caso o resultado obtido não coincidir com as saídas esperadas, o conjunto de casos de teste revelou a presença de defeitos. A Figura 2 representa a configuração do cenário descrito.

Figura 2 – Cenário típico da atividade de teste



Fonte: Delamaro, Maldonado e Jino (2016).

Considerando a Figura 2, percebe-se que foi selecionado um conjunto de valores específicos para serem utilizados na execução do programa em teste, assim, nem todos os valores foram selecionados. De acordo com Delamaro, Maldonado e Jino (2016), seria

ideal utilizar todos os valores possíveis para garantir que o programa não contém defeitos. No entanto, por causa da grande quantidade de valores, esta ação é, em geral, infactível. Com tal conhecimento, pesquisas foram conduzidas e chegaram a formas que possibilitam o uso de um subconjunto reduzido de valores para execução de um programa ou de um software. Essas formas visam a redução do custo do teste e estabelecem estratégias para o aumento da eficácia da atividade (SOUZA *et al.*, 2015) e são reconhecidas como critérios de teste. Por sua vez, esses critérios são agrupados em técnicas de teste de software que se diferenciam pela origem da informação utilizada para estabelecer o subconjunto reduzido de valores e consequentemente o conjunto de casos de teste (MALDONADO *et al.*, 1998) e para fazer a avaliação da adequação desse conjunto (FRANKL; WEYUKER, 2000).

Basicamente, os critérios de teste são agrupados em três técnicas principais, a saber: teste funcional, teste estrutural e teste baseado em defeitos. Elas são fundamentais para a seleção de casos de teste e buscam garantir a cobertura de determinadas informações do programa (MELO, 2012). Nesse sentido, devido à capacidade delas de cobrir determinadas partes, cada uma delas traz certo tipo de vantagem. Portanto, elas devem ser interpretadas como complementares (NAKAGAWA, 2006). Cada técnica de teste de software utiliza um tipo de informação para selecionar casos de teste que buscam identificar o maior número de defeitos existentes no software. Nas próximas subseções serão apresentadas as principais técnicas e seus respectivos critérios de teste.

2.2.1 Técnicas e Critérios de Teste

Teste Funcional

A técnica de teste funcional utiliza informações da especificação do programa para a geração de casos de teste, por esse motivo, o teste funcional é um tipo de teste que é normalmente conhecido como Teste Caixa-Preta. Em decorrência de não se conhecer o código do programa, os casos de teste são projetados a partir da especificação do produto de software. Nesse sentido, a partir da especificação, é definido o conjunto de valores que podem ser utilizados para executar o programa em teste, bem como conjunto de casos de teste. Em seguida, o programa é testado e o conjunto de resultados produzidos pelo programa é comparado com a especificação do programa.

Para a condução sistemática do teste funcional, alguns critérios foram definidos: Particionamento em Classes de Equivalência e Análise do Valor Limite. Cada um desses critérios será apresentado a seguir:

- **Particionamento em Classes de Equivalência**: O particionamento em classes de equivalência parte da premissa de tornar a quantidade de dados de entrada finita e viável de ser testada, oferecendo condições para que um subconjunto de dados seja definido, que em tese, deve ser capaz de revelar erros que seriam revelados

somente com o Teste Exaustivo (DELAMARO, 1997). Para tanto, esse critério divide o domínio de entrada em classes de equivalência (válidas e inválidas), a partir das condições de entrada do programa ou software, considerando a especificação do mesmo (FABBRI *et al.*, 2016). Desse modo, os casos de teste são provenientes das classes de equivalência, uma vez que ao definir um valor que faz parte de diversas classes de equivalência, diminui-se a quantidade de teste que deveria ser feita, reduzindo o esforço necessário para se testar esse software (SOUZA, 2000). Em tese, um elemento que faz parte de uma classe, representa toda a sua classe.

- **Análise do Valor Limite**: este critério complementa o critério Particionamento em Classes de Equivalência, buscando testar os limites de cada classe de equivalência. Assim, esse critério seleciona os valores que estão exatamente sobre ou imediatamente acima ou abaixo dos limites das classes de equivalência (FABBRI *et al.*, 2016). Portanto, em vez de selecionar valores aleatórios das classes de equivalência, conforme ocorre no critério Particionamento em Classes de Equivalência, esses valores são extraídos dos limites das classes.

Além dos critérios apresentados acima, existem outros que também apoiam o teste funcional, a saber: Teste Funcional Sistemático, Grafo Causa-Efeito, *Error-Guessing*, *Syntax Testing*, *State Transition Testing* e *Graph Matrix* (FABBRI *et al.*, 2016). Como todos os critérios são derivados da especificação, eles podem ser aplicados independentemente do paradigma de programação. No entanto, é importante salientar que a eficiência dos critérios do teste funcional depende da qualidade da especificação. Além disso, ele ignora propriedades importantes que fazem parte da implementação e não são descritas no projeto de software (HOWDEN, 1986), especialmente relacionadas ao código-fonte. Nesse sentido, recomenda-se que o software em teste, seja testado com critérios de diferentes técnicas (NAKAGAWA, 2006). Nesse sentido, o uso de critérios da técnica de teste funcional com a técnica de teste estrutural é um meio para garantir que um programa foi suficientemente testado. Com base nisso, a próxima subseção visa apresentar o teste estrutural de software.

Teste Estrutural

A técnica de teste estrutural baseia-se na estrutura interna do programa ou software (CRESPO, 1997). Nesse sentido, a pessoa que conduz a atividade de teste deve levar em consideração os dados de entrada, a semântica e sintaxe do programa (nessa atividade características da linguagem de programação devem ser consideradas), o programa em execução e os dados de saída para elaborar os casos de teste. Assim, enquanto se faz a analogia do teste funcional a uma “Caixa-Preta”, pode-se analogamente relacionar o teste estrutural com uma “Caixa-Branca”, em contradição ao funcional.

Nesse tipo de critério, os caminhos lógicos do programa ou software em teste são testados. Assim, os critérios da técnica oferecem condições para demonstrar erros presentes na semântica do programa, como problemas associados à sequência lógica de passos que o computador executará fielmente para que se obtenha os resultados (BARBOSA *et al.*, 2016). Essa sequência, por sua vez, será constituída de estruturas básicas (*e.g.*, sequência, seleção e de interação condicional), variáveis e objetos, classes, métodos e funções (dependendo do paradigma de programação) que serão testadas (PRADO, 2009). Por consequência de a técnica envolver a análise sobre diferentes tipos de conceitos e elementos do programa, os critérios da técnica Estrutural de software se subdividem em quatro categorias de critérios principais, são elas: (i) critérios baseados na complexidade; (ii) critérios baseados em fluxo de controle; (iii) critérios baseados em fluxo de dados; e, por fim, (iv) critérios baseados na cobertura (BARBOSA *et al.*, 2016). Uma visão geral das principais categorias (critérios baseados em fluxo de controle e critérios baseados em fluxo de dados) será apresentada a seguir:

Critérios baseados em fluxo de controle: os critérios baseados em fluxo de controle são constituídos por uma família de critérios de teste que se baseiam nas estruturas de controle do código para derivar os requisitos de teste (BARBOSA *et al.*, 2016). Dentre esses critérios destacam-se: Teste de Comandos, Teste de Arcos e Teste de Condições (CRESPO, 1997).

Teste de Comandos: Esse critério estabelece que o testador deve construir dados de teste que contemplem a execução de cada comando do programa, ao menos uma vez (CRESPO, 1997). O critério também é conhecido como "Todos-Nós" (BARBOSA *et al.*, 2016).

Teste de Arcos: Esse critério é conhecido por mais duas diferentes nomenclatura, "Todas-Arestas" (BARBOSA *et al.*, 2016) e "Teste de Ramos" (CRESPO, 1997). Ele busca orientar o testador a construir dados de teste de tal maneira que cada desvio do fluxo do programa (pode-se interpretar como: desvio condicional simples, desvio condicional composto, dentre outas formas) seja testado pelo menos uma vez (CRESPO, 1997).

Teste de Condições: Tem como objetivo direcionar o testador a construir dados de teste de tal maneira que cada condição dentro de uma decisão (pode-se interpretar como estruturas de decisão) seja testa ao menos uma vez (CRESPO, 1997).

Critérios baseados em fluxo de dados: As famílias que constituem os critérios baseados em fluxos de dados analisam o fluxo de dados dos programas ou software em teste como fonte de informação para derivar os requisitos de teste, avaliando a definição e o uso das variáveis ao longo do programa (RAPPS; WEYUKER, 1985). Por definição de

variável entende-se por: uma variável é definida quando um valor é armazenado em uma posição de memória (CRESPO, 1997). Assim, diferente do critério baseado em fluxo de controle, esse critério foca nas variáveis do programa ou software em teste. Os principais critérios são apresentados a seguir:

Todas-Definições: Estabelece que cada definição de variável seja executada ao menos por um uso dessa variável, podendo ser por um uso computacional, que é o uso em uma computação ou cálculo (também conhecido como *c-usos*[1]) ou por um uso de variável em uma expressão condicional, associada a comandos de desvio ou de repetição (também conhecido como *p-usos*[2]).

Todos-Usos: estabelece que todas as associações entre uma definição de variável e todos os usos da mesma (*c-usos* e *p-usos*) sejam executadas pelos casos de teste, por meio de pelo menos um caminho em que a variável não seja redefinida.

Todos-Du-Caminhos: estabelece que todas as associações entre uma definição de variável e todos os usos da mesma (*c-usos* e *p-usos*) sejam executadas pelo conjunto de teste, por todos os caminhos livre de definição e laços que sejam capazes de cobrir essa associação.

No decorrer desta subseção foram discutidas os critérios e as famílias de critérios que apoiam a aplicação do teste de software. Os critérios de teste estrutural apoiam aspectos não cobertos com o teste funcional. É importante frisar que a literatura apresenta outras técnicas de teste, que consideram outros elementos que são importantes. Dentre essas técnicas, o Teste Baseado em Defeitos baseia-se na construção de casos de teste que consegue revelar defeitos típicos cometidos por programadores. Nesse sentido, é uma técnica que podem complementar o teste estrutural. Assim, a próxima subseção visa apresentar a técnica e o seu principal critério.

Teste Baseado em Defeitos

A técnica baseada em defeito é constituída por critérios que buscam definir requisitos de testes levando em consideração os defeitos típicos do processo de implementação do software (DEMILLO; LIPTON; SAYWARD, 1980). Essa técnica coloca em ênfase os defeitos que normalmente são cometidos pelo programador durante o processo de implementação do código. Nela, modelos de defeitos são utilizados para criar hipóteses sobre defeitos que podem estar presentes no programa ou software e elaborar ou avaliar o conjunto de casos de teste com base na capacidade que os mesmos possuem em revelar os defeitos que foram modelados por meio das hipóteses (ANDRADE, 2016). Um dos principais critérios associados à técnica é o teste de mutação, que será apresentado a seguir.

[1] Por exemplo: a=b+b;
[2] Por exemplo: if(a>b);

- **Teste de Mutação**: O critério de teste de mutação - ou análise de mutantes ou ainda mutação forte - tem como principal objetivo mensurar o grau de adequação de um conjunto de casos de teste (DEMILLO; LIPTON; SAYWARD, 1980). Para tanto, versões alternativas do programa original são geradas, normalmente com auxílio de ferramentas de apoio. Essas versões alternativas contém pequenas modificações sintáticas que representam os possíveis defeitos que poderiam estar presentes no programa (DELAMARO *et al.*, 2016). Esse programas alternativos são denominados mutantes. Assim, devem ser elaborados casos de teste que consigam revelar, por intermédio da execução do programa, as diferentes de comportamento existente entre o programa original e os programas mutantes (DELAMARO, 1993; DELAMARO, 1997). Com isso, é possível demonstrar que o programa ou software em teste não possui determinados tipos de defeito.

  A aplicação do teste de mutação está diretamente associada ao paradigma e linguagem de programação, uma vez que os programas mutantes são criados a partir de operadores de mutação[3], os quais são específicos para uma linguagem. Assim, Os operadores de mutação dependem da linguagem de programação, por exemplo, a linguagem C tem 75 operadores, já a linguagem FORTRAN possui apenas 22 operadores. De acordo Delamaro (1993) o conjunto de 75 operadores de mutação da linguagem C são divididos em quatro classes, a saber: mutação de comandos, mutação de operadores, mutação de variáveis e mutação de constantes. Considerando que cada linguagem possui um número especifico de operadores, os mesmos também são classificados em categorias.

No decorrer desta subseção foram apresentadas algumas conceituações sobre o teste baseado em defeitos. Percebe-se que os objetivos dos critérios dessa técnica consistem em oferecer subsídios ao testador em elaborar com eficácia um conjunto de casos de teste com maiores condições de revelar defeitos.

Considerando-se que um dos principais objetivos deste trabalho é oferecer subsídios a formação de recursos humanos em teste de software, a próxima seção discute o estado da arte relacionado às contribuições para com o ensino de teste de software. Nesta perspectiva, apresenta uma síntese das principais pesquisas que estão sendo conduzidas e que serviram como base para motivar a realização do presente trabalho de mestrado.

2.2.2 Ensino de Teste de Software

Muitas pesquisas no contexto de ensino de teste de software discorrem sobre as dificuldades de ensinar e motivar os alunos de computação sobre a importância de realizar

[3] São regras que definem as alterações que dever ser feitas no programa em teste para criar os programas mutantes (SOUZA, 2000).

um bom teste de software (BARBOSA; MALDONADO, 2011; CLARKE *et al.*, 2011; SOSKA; MOTTOK; WOLFF, 2016; ROJAS *et al.*, 2017). Na seção anterior foi discutido sobre a importância da atividade de teste de software. Além disso, no Capítulo 1 desta dissertação foram mencionados alguns estudos que investigam o apoio ao ensino de teste de software, por meio de modelos pedagógicos com viés às práticas de ensino alternativas às tradicionais. Todas essas pesquisas são construídas levando em consideração o déficit de profissionais existente em teste de software, dado que muitos fatores contribuem com a formação não adequada dos estudantes de computação, e a indústria de desenvolvimento de software vem tendo um crescimento significativo nos últimos anos, o que aumenta a demanda por pessoal capacitado, com boas habilidades na área.

Existe uma vertente de pesquisa que apresenta indícios que é necessário modificar as estratégias de ensino, por intermédio de métodos de ensino e/ou modelos pedagógicos que promovam o aluno ao centro da aprendizagem, deixando a responsabilidade em aprender por conta do aluno (SOSKA; MOTTOK; WOLFF, 2016). No entanto, as estratégias e abordagens emergentes para aprimorar as habilidades dos futuros profissionais de teste de software, estão sendo pouco investigadas (SMITH *et al.*, 2012), em comparação à pesquisa sobre o estabelecimento de mecanismos de apoio (VALLE; BARBOSA; MALDONADO, 2015). Apesar da prematuridade dos esforços, já existem algumas iniciativas. Os estudos de Figueredo *et al.* (2011) e Smith *et al.* (2012), por exemplo, relatam experiências em sala de aula sobre a aplicação de abordagens de ensino que oportunizam uma participação mais ativa dos alunos.

Figueredo *et al.* (2011) apontaram alguns problemas no âmbito de teste de software: (i) o ensino oferecido pelas instituições acadêmicas é geralmente direcionado à apresentação de conceitos e técnicas de teste por meio de um método tradicional que é focado no professor; (ii) as atividades executadas pelos alunos são suscetíveis de ficarem distantes das complexas que os futuros profissionais enfrentarão nos problemas do mundo real; (iii) os estudantes normalmente são avaliados pelos professores no que se refere aos aspectos teóricos do conhecimento da área, deixando de considerar o conhecimento para aplicação prática desses conceitos. Diante disso, mencionaram sobre a necessidade em dar ênfase em um modelo pedagógico alternativo ao método tradicional de ensino.

Figueredo *et al.* (2011) investigaram em sua pesquisa o modelo PBL (*Problem Based Learning*), no qual o estudante é desafiado a participar ativamente do processo de aprendizagem, responsabilizando-se pela sua aprendizagem. Os autores também argumentam que o modelo possibilita aos alunos o trabalho em equipe, em que os mesmos devem se organizar para solucionar problemas. Adicionalmente, o modelo encoraja o aluno a realizar atividades que possibilitam o desenvolvimento de habilidades e aprimoramento de atitudes. Para uso da abordagem, os autores definem as habilidades e os conhecimentos que os alunos deveriam desenvolver e adquirir, selecionando projetos do mundo real,

envolvendo a participação de clientes/*stakeholders*. Com base nisso, os alunos precisarão participar desses projetos, trabalhar em equipe, interagir com *stakeholders*, tendo aulas com viés ao aprendizado teórico-prático.

No estudo de Smith *et al.* (2012) os autores apresentam uma abordagem diferenciada para ensinar teste de software na educação formal em computação. Essa abordagem é a revisão por pares (do inglês: *peer review*), também reconhecida como teste por pares (do inglês: *peer testing*). Apesar de ser diferente da tradicional, ela não é uma novidade na área de teste de software, uma vez que Clark (2004) já havia publicado um estudo sobre a possibilidade do uso da mesma no contexto de ensino de teste de software. Smith *et al.* (2012) descrevem uma experiência sobre a incorporação de teste por pares com estudantes da disciplina de estrutura de dados, visando demonstrar a importância da atividade, apresentando o teste como uma atividade divertida e competitiva.

A abordagem desenvolvida por Smith *et al.* (2012) busca incentivar os estudantes a encontrar defeitos em estruturas de dados ou casos de teste desenvolvidos por seus pares. A prática cria um ambiente de competição saudável para os estudantes, ao passo que os alunos ficam estimulados a encontrar problemas nos trabalhos desenvolvidos por seus colegas, aqueles que desenvolvem os casos de teste ou que codificam as estruturas ficam mais atento para não deixar o programa sujeito a conter defeitos. Nesse estudo, Smith *et al.* (2012) também abordam a inciativa do ensino e teste em conjunto com programação. Um estudo semelhante sobre revisão por pares foi conduzido e apresentado recentemente por (BARBOSA *et al.*, 2017), neste caso, o trabalho é específico para estudantes da disciplina de teste de software.

Ao longo desta seção foram foram discorridos estudos sobre o uso de diferentes iniciativas que buscam fornecer condições mais propícias para a aprendizagem dos alunos. No Capítulo 1 também havia sido mencionado uma investigação socializada pelo autor desta pesquisa acerca do uso do modelo *flipped classroom* para apoiar o ensino de teste de software, divulgados em Paschoal, Silva e Souza (2017). O modelo intitulado *flipped classroom* é considerado uma metodologia ativa de aprendizagem, que coloca o estudante no centro da aprendizagem e ao mesmo tempo é um modelo sustentado de ensino que se apoia pelas novas tecnologias da informação e comunicação. Nessa perspectiva, a próximo seção visa apresentar o modelo *flipped classroom*.

2.3 Flipped Classroom

Flipped classroom desde o seu surgimento vem sendo explorado como um modelo pedagógico que visa apoiar a aprendizagem ativa. Nesse modelo, o professor deixa seu papel de transmissor de conhecimentos de lado e passa a ser um orientador/mediador do aluno. Os estudantes, por sua vez, passam a ser responsáveis por estudar previamente o

conteúdo antes da aula, e na sala de aula utilizam o tempo para exercitar seus conhecimentos, realizando mais atividades práticas (*e.g.* exercícios, listas, projetos, discussões sobre a temática e exercícios, dentre outros). Ele pode ser considerado um modelo de ensino alternativo para o tradicional, posto que pode ser entendido como uma inversão organizacional do método de ensino tradicional.

No método de ensino tradicional, em sala de aula os estudantes fazem leituras sobre o conteúdo e assistem a apresentações feitas pelos professores, finalizando as atividades de fixação de conceitos e teorias fora da sala de aula (VALENZUELA *et al.*, 2013). Assim, a maior parte do tempo "gasto" em sala de aula pelos estudantes é destinado a assistir ao professor expondo o conteúdo. Essa prática de ensino, apesar de ainda ser muito utilizada, faz com que o professor esteja no centro da aprendizagem. Além disso, como o tempo é "gasto" com exposição de conteúdo, pouco tempo sobra para os estudantes trabalharem com atividades práticas de fixação. Por falta de tempo, os estudantes concluem em suas casas os trabalhos e exercícios de fixação.

No modelo tradicional, a prática de conceitos e teorias é pouco exercitada em sala de aula e existem poucas oportunidades para os estudantes solucionarem suas dúvidas. No entanto, a maioria das dúvidas dos estudantes surge nos momentos em que os mesmos aplicam os conceitos vistos em aula (BERGMANN; SAMS, 2012). Dessa forma, muitas vezes os alunos não conseguem solucionar as dúvidas que surgem no momento em que os mesmos tentam solucionar os exercícios propostos pelo professor. Em sala de aula, o professor dedica pouco tempo para apresentar as resoluções dos exercícios apresentados, pois necessita apresentar um novo conteúdo e progredir a proposta curricular do curso.

As pesquisas sobre *flipped classroom* destacam a importância do planejamento e documentação da estratégia (FASSBINDER *et al.*, 2014). Existem alguns pesquisadores que vêm construindo recursos para apoiar o planejamento da inversão da sala de aula, com vistas a aumentar o desejo dos professores em utilizar a abordagem. Bergmann e Sams (2012) discorrem sobre como planejar a aula, apresentando algumas estratégias que podem ser adotadas. Fassbinder *et al.* (2014), por sua vez, apresentam mecanismos que ajudam na definição e implementação do modelo *flipped classroom*. Schmitz (2016) também colaboram, compilando e sintetizando algumas ideias da literatura para criar algumas dicas para uso do modelo. Gilboy, Heinerichs e Pazzaglia (2015), no que lhe diz respeito, estruturam um *template* que pode ser utilizado no planejamento de incorporação do modelo em aulas. Nesse sentido, o número de estudos que se concentram em contribuir com a adoção do modelo é amplo. A seguir será apresenta uma visão geral sobre a implementação do modelo *flipped classroom*.

2.3.1 Implementação de Flipped Classroom

Bergmann e Sams (2012) discorrem que o primeiro passo para o professor adotar o modelo é definir o objetivo da aula. Posteriormente, deve verificar quais os mecanismos de apoio que serão utilizados para atender o objetivo educacional da aula. O vídeo instrucional é uma das ferramentas mais adotadas no modelo, no entanto, é necessário analisar de forma cuidadosa se o vídeo é a ferramenta adequada. Se o professor optar por adotar vídeo, ele precisa ponderar se criará seus próprios vídeos ou utilizará vídeos de outros professores. Com o advento do YouTube e de outros repositórios de vídeos educacionais, a quantidade de vídeos instrucionais com boa qualidade está em ascensão. Assim, muitos vídeos produzidos por terceiros, disponíveis nesses repositórios, podem ser usados na adoção do modelo.

Além dos vídeos disponíveis no YouTube, Fassbinder *et al.* (2015) descrevem que os vídeos disponíveis em MOOCs (*Massive Open Online Courses*) podem apoiar a adoção do modelo. Entretanto, conforme já mencionado, *flipped classroom* não se constitui apenas de vídeos instrucionais. Se o professor optar por ser autor dos vídeos, Bergmann e Sams (2012) apresentam algumas sugestões. Com a adoção do modelo, eminentemente ocorrerá um acréscimo de tempo disponível para atividades práticas em sala de aula. Esse ganho de tempo deverá ser avaliado e explorado da melhor maneira possível, em atividades úteis para compreensão com mais profundidade as complexidades dos conceito, como a resolução de projetos, condução de experimentos, debates e discussões. Inclusive, atividades que exploram a geração de conteúdo pelo próprio aluno.

Por ser um modelo pedagógico, *flipped classroom* precisa ser implementado levando em consideração os objetivos e competências que precisam ser atingidas pelos alunos (TEIXEIRA, 2013). Nesse sentido, teorias vêm sendo utilizadas para oferecer suporte ao planejamento pedagógico. Ferraz e Belhot (2010) citam a taxonomia de Bloom. Essa taxonomia pode ser utilizada por professores com o intuito de estabelecer os diferentes objetivos de aprendizagem. Nesse sentido, o docente deve elaborar um plano com o intuito de contemplar as seis categorias do domínio cognitivo (recordar, compreender, aplicar, analisar, avaliar e criar), nos três momentos proporcionados pelo modelo: (i) antes da aula (do inglês: *before class*), (ii) durante a aula (do inglês: *in class*), (iii) depois da aula (do inglês: *after class*).

Com o objetivo de auxiliar na implementação do modelo *flipped classroom* em um curso de Nutrição, Gilboy, Heinerichs e Pazzaglia (2015) durante a descrição da intervenção, construíram um *template* para ajudar na organização dos três momentos proporcionados pelo modelo. A Figura 3 apresenta uma versão do *template* traduzida pelo autor desta pesquisa. No campo Recursos, os autores recomendam que sejam especificados os recursos que serão utilizados como *websites*, artigos, vídeos, dentre outros. No espaço Atividades os autores sugerem que o mesmo seja utilizado para especificar as atividades

que os alunos deverão realizar, como discussão, fazer anotações, leituras, assistir a vídeos, palestras, dentre outros. Por fim, no campo Avaliação, os autores aconselham que deve ser utilizado para evidenciar as atividades que o professor irá propor a turma.

Figura 3 – Exemplo de um *template* de planejamento de *Flipped Classroom*

Semana 1	Nome do Tópico:			
	Objetivos de aprendizagem para o tópico: ❖ <<lista>>			
	Recursos: ❖ <<lista>>			
		Antes da aula	Na aula	Depois da aula
	Atividades	<<descrição>>	<<descrição>>	<<descrição>>
	Avaliação	<<descrição>>	<<descrição>>	<<descrição>>

Fonte: Adaptada de Gilboy, Heinerichs e Pazzaglia (2015).

Na literatura da área é possível encontrar muitos outros estudos que apresentam indicações de quais procedimentos o professor deve seguir para adotar o modelo *flipped classroom*, especialmente associados ao seu planejamento. No decorrer desta seção algumas iniciativas foram apresentadas. Muitas outras iniciativas vêm sendo estabelecidas. As contribuições são propostas porque o modelo vem recebendo muita atenção por parte da comunidade acadêmica, em virtude de suas vantagens. Além disso, há também algumas iniciativas que buscam minimizar as desvantagens que vem sendo descobertas. A próxima seção visa apresentar uma discussão sobre vantagens e desvantagens que vêm sendo descritas na literatura.

2.3.2 Vantagens e desvantagens de Flipped Classroom

Mesmo sendo um modelo recente, que vem sendo investigado aos poucos em algumas áreas, já existem estudos que apresentam as vantagens e desvantagens ao se fazer uso do modelo. No decorrer desta seção, serão apresentadas algumas vantagens que são evidenciadas pela literatura bem como as desvantagens/limitações percebidas por professores sobre uso do mesmo. Também serão apresentadas as vantagens e desvantagens percebidas por estudantes que aprendem com o modelo e possuem histórico de estudarem com o método tradicional de ensino.

No contexto de computação, por exemplo, Zhamanov e Sakhiyeva (2015) ao utilizarem o modelo no contexto de ensino de Redes de Computadores, mencionam os benefícios do modelo e suas desvantagens de uso. Sobre os benefícios, em especial, discorrem que os professores passam a ter mais interação face a face com os estudantes e os alunos passam

a reforçar seus conhecimentos ao compartilharem os mesmos com seus pares durante o exercício em sala de aula. Já sobre as desvantagens, discursam que os estudantes desinteressados não assistem às aulas e acabam perdendo o conteúdo; a inversão da sala de aula exige tempo extra dos professores para preparar os materiais didáticos.

Fulton (2012) também listou algumas vantagens do modelo, destacado principalmente que os alunos aprendem no seu próprio ritmo; os professores consegue ter uma visão melhor sobre as reais dificuldades dos alunos por meio das atividades realizadas em sala de aula; professores que observam os resultados do modelo, percebem níveis maiores de realização, interesse e engajamento dos estudantes. Herreid e Schiller (2013) complementam o estudo de Fulton (2012), citando benefícios adicionais: os alunos conseguem tempo para trabalhar com equipamentos que só podem ser acessados em sala de aula; os alunos ficam envolvidos ativamente no processo de aprendizagem.

Além de complementar os benefícios citados por Fulton (2012), Herreid e Schiller (2013) mencionam que os alunos podem ser inicialmente resistentes e os trabalhos de casa devem ser cuidadosamente preparados pelos professores. Realçando o texto de Fulton (2012) e Herreid e Schiller (2013), Syam (2014) também descrevem sobre alguns problemas que podem ser enfrentados pelo professor durante a implementação do modelo, são eles: os alunos podem estar despreparados para trabalhar com o modelo e consequentemente poderão ser resistentes; fazer ou encontrar um vídeo instrucional de boa qualidade é difícil; os vídeos produzidos por terceiros podem não coincidir com os objetivos de aprendizagem; a autoria de vídeos requer uma dedicação significativa do tempo do professor.

Outro estudo que referencia Fulton (2012) e Herreid e Schiller (2013) e aponta alguns benefícios e desafios, é o trabalho de Wood (2016). No que se refere aos benefícios promovidos destacam-se: as preeleções sobre o conteúdo podem ser visualizadas quantas vezes forem necessárias; o tempo de contato em sala de aula passa a ser mais significativo e útil. Sobre as limitações, salientam que alguns alunos sentem dificuldade com o modelo, pois não conseguem fazer perguntas em sala de aula sobre suas dúvidas; o uso do modelo em turmas grandes dificulta o gerenciamento das mesmas.

No decorrer desta seção foram apresentadas algumas vantagens e desvantagens do uso do modelo, sobre a ótica do professor. No estudo de Ramírez, Hinojosa e Rodríguez (2014) os autores buscaram identificar a percepção de estudantes de cursos de engenharia sobre as vantagens e desvantagens do modelo. Os resultados revelam que os estudantes destacam como vantagens a flexibilidade em aprender por meio de vídeos; a segurança por terem acesso a uma maneira mais fácil de formação; a motivação. Dentre as desvantagens mencionadas pelos alunos estão: problemas técnicos associados à Internet, software, dentre outros; falta de *feedback* instantâneo no momento em que os alunos assistem os vídeos, isto é, quando os alunos tem dúvida sobre o conteúdo, eles não conseguem solucioná-las imediatamente; vídeos instrucionais longos.

A falta de auxílio nos momentos em que eles estão estudando por conta própria o conteúdo recomendado pelo professor é uma notável limitação. Com o surgimento de dúvidas, os alunos podem não conseguir continuar os estudos ou, mesmo conseguindo continuar, podem ficar suscetíveis a cometerem enganos e ficarem prejudicados. Tradicionalmente, no modelo *flipped classroom* essas dúvidas devem ser levadas para sala de aula, sendo solucionadas em momentos dedicados. Mesmo existindo esse espaço, pode ser que nem todos os alunos consigam divulgar suas dúvidas para o professor, pois a turma pode ser muito grande. Os estudantes podem ficar intimidados em demonstrar que ficaram com dúvidas, principalmente aqueles tímidos.

Até o presente momento, os mecanismos de apoio utilizados no modelo *flipped classroom* não contemplam a tutoria artificial, que pode ser realizada por agentes conversacionais.Esses programas, no contexto educacional, podem oferecer diversos benefícios para o modelo, podendo ser utilizados para apoiar o momento "antes da aula" e atividades assíncronas pré-classe, solucionando as dúvidas iniciais dos estudantes. Buscando explorar esse tipo de sistema de software, a próxima seção apresentará uma visão geral sobre agentes conversacionais.

2.4 Agentes Conversacionais

Os agentes conversacionais são programas de computador que são capazes de interagir em linguagem natural com seres humanos (DERYUGINA, 2010), sobre um determinado domínio ou tópico (WOUDENBERG, 2014). Tais sistemas, desde o seu surgimento por volta dos anos 60, oferecem apoio para a comunicação homem-máquina se tornar mais próxima à linguagem natural, realizada pelos seres humanos (MANFIO; MORENO, 2015). Nesse sentido, podem ser considerados como os primeiros programas a serem desenvolvidos com o intuito de se comunicar como um humano. Para tanto, ao interagirem com uma pessoa, eles precisam interpretar e processar as mensagens transmitidas pelo sujeito, por canais de comunicação, buscando uma possível resposta em uma base de conhecimento, visando responder as mensagens, atendendo com satisfação à solicitação realizada.

A terminologia agente conversacional é amplamente utilizada pela literatura para designar sistemas de software que simulam a conversa com seres humanos, utilizando linguagem natural, especialmente em trabalhos brasileiros, a saber: Paschoal, Oliveira e Chicon (2018), Aguiar, Tarouco e Reategui (2014), Santos (2009). Mesmo assim, é possível encontrar pesquisas sobre tais programas utilizando outros termos, tais como: *chatterbots*, *chatbots*, *chat bots*, *bots*, agentes inteligentes, agentes conversacionais incorporados, agentes virtuais, assistentes virtuais, sistemas de diálogo e agentes pedagógicos. É importante salientar que algumas dessas terminologias devem ser utilizadas considerando as características do agente conversacional em questão. Além disso, reconhece-se que não existe um

consenso sobre a adequação de uso de tais terminologias (SANTOS, 2009), e pesquisadores vêm tentando explicar como fazer uso das mesmas.

Devido à crescente expansão da Internet, os agentes conversacionais foram sendo desenvolvidos e utilizados para os mais diversificados ambientes, com diferentes propósitos, pois podem trazer benefícios para diferentes áreas. Radziwill e Benton (2017) mencionam a área comercial, uma vez que possibilitam o atendimento ao cliente, oferecendo agilmente as respostas. Outro campo que vem recebendo incentivos à pesquisa é o educacional (AGUIAR; TAROUCO; REATEGUI, 2014). Agentes conversacionais para apoiar os processos de ensino e aprendizagem são construídos para diferentes áreas de ensino, em especial para a área da computação, para apoiar o ensino de Programação (BIGHAM *et al.*, 2008; WIGGINS *et al.*, 2017), Engenharia de Software (PASCHOAL; CHICON; FALKEMBACH, 2017; PASCHOAL; OLIVEIRA; CHICON, 2018), Redes de Computadores (LEONHARDT *et al.*, 2007; HERPICH *et al.*, 2016), dentre outros.

Agentes conversacionais para ser utilizados na temática educacional, podem contribuir de diferentes formas, tais como: solucionar dúvidas (GRAESSER *et al.*, 2017); fazer com que os alunos atentem-se a informações relevantes (CRAIG; SCHROEDER, 2017); oferecer informações instrucionais (BICKMORE *et al.*, 2010); oferecer *feedback* imediato e elaborado dado o seu potencial para ajudar os alunos a monitorar sua própria aprendizagem (GRAESSER *et al.*, 2017); aumentar a motivação do usuário por meio de discurso motivacional (CRAIG; SCHROEDER, 2017); oferecer comportamentos afetivos que impactam positivamente nos estados afetivos dos alunos (LIN *et al.*, 2013); dentre outras. Assim, os agentes podem ser utilizados para apoiar o ensino de diferentes populações (estudante do ensino médio, estudantes universitários, alfabetização de adultos, dentre outros) (GRAESSER *et al.*, 2017), nas diferentes modalidade de ensino (presencial ou a distância).

Existem muitas pesquisas científicas que relatam sobre o desenvolvimento, utilização e relevância de agentes conversacionais no ensino de computação, com o intuito de complementar de forma mais eficaz processos de ensino e aprendizagem. Tendo em vista o objetivo desta dissertação de mestrado, buscou-se alguns estudos que descrevem sobre a incorporação desses mecanismos de apoio às diferentes subáreas da computação. Na próxima seção, serão apresentados alguns agentes conversacionais estabelecidos para ser utilizados em ambientes de aprendizagem, com o propósito de apoiar o ensino de conteúdos de sub-áreas de computação.

2.4.1 Agentes no Ensino de Computação

No estudo de Silveira e Thiry (2010) foi desenvolvido um agente conversacional para apoiar o ensino de Medição de Software, dado que a forma que ela é ensinada é vista como difícil de ser aprendida e os cursos de graduação em computação não dedi-

cam tempo suficiente sobre esse assunto. O agente conversacional foi implementado na linguagem AIML (*Artificial Intelligence Modelling Language*). A base de agente conversacional manter uma conversa com um humano, em especial base de conhecimento com saudações, base de conhecimento sobre o agente conversacional (perfil do agente), base de conhecimento com assuntos sobre conversas do dia a dia, e base de conhecimento sobre o domínio específico. A base de domínio específico, por sua vez, foi implementada a partir de conhecimentos adquiridos pelos autores durante revisões bibliográficas.

Lemos (2011) desenvolveu um agente conversacional que tem papel de tutor virtual no contexto de ensino de Algoritmos. Esse agente é denominado Cocorote, e também foi implementado em AIML. Cocorote tem capacidade de expressar suas próprias emoções (alegria, tristeza, surpresa, raiva, desgosto e medo) alterando sua representação visual. Durante a interação com o aluno, Cocorote modifica suas expressões por meio de alterações de sua representação visual. Essas modificações de representação visual foram especificadas previamente a partir de estudos com usuário que interagiram com as bases de conhecimento, eles definiram durante a conversa quais as emoções que Cocorote deveria expressar com base nos pares conversacionais. Os autores tiveram que utilizar comandos da linguagem PHP (*Hypertext Preprocessor*) para apoiar a modificação das expressões emitidas pelo *chatterbot*. A Figura 4 apresenta a interface principal de Cocorote.

Figura 4 – Exemplo de interação entre Cocorote e aluno

Fonte: Lemos (2011).

O estudo de Bos (2014) apresenta o desenvolvimento de um agente conversacional para um museu sobre a História da Computação. O museu foi desenvolvido com o servidor de mundos virtuais OpenSimulator[4] é destinado a estudantes de graduação em computação, mais especificamente alunos matriculados em disciplinas introdutórias de

[4] Mais informações disponíveis em: <www.opensimulator.org/>.

computação. O agente é denominado AGIMC (Agente Inteligente Conversacional como Guia em um Museu Virtual 3D da Computação) visa orientar o aluno durante a exploração no museu. Para tanto, foi desenvolvimento com a linguagem AIML. No museu, o agente tem o papel de guiar o estudante, contendo conhecimentos sobre aspectos técnicos e culturais da área, do passado e do presente, sobre a trajetória da computação nas diferentes décadas de sua evolução.

ELAI é um agente conversacional desenvolvido por Herpich *et al.* (2016) que possibilita apoiar os estudantes de Redes de Computadores durante a utilização de um ambiente de aprendizagem imersivo que prove acesso ao conteúdo considerando informações contextuais (tecnologia, estilos cognitivos, nível de expertise e qualidade de contexto). Esse *chatterbot* é representado por um NPC (*Non Player Character*) e tem suas bases de conhecimentos desenvolvidas em AIML. ELAI tem parte de suas bases de conhecimento em português e parte em inglês, que foram disponibilizadas na plataforma Pandorabots[5], um interpretador da linguagem. A Figura 5 representa ELAI interagindo com estudantes.

Figura 5 – ELAI interagindo com o aluno

Fonte: Herpich *et al.* (2016).

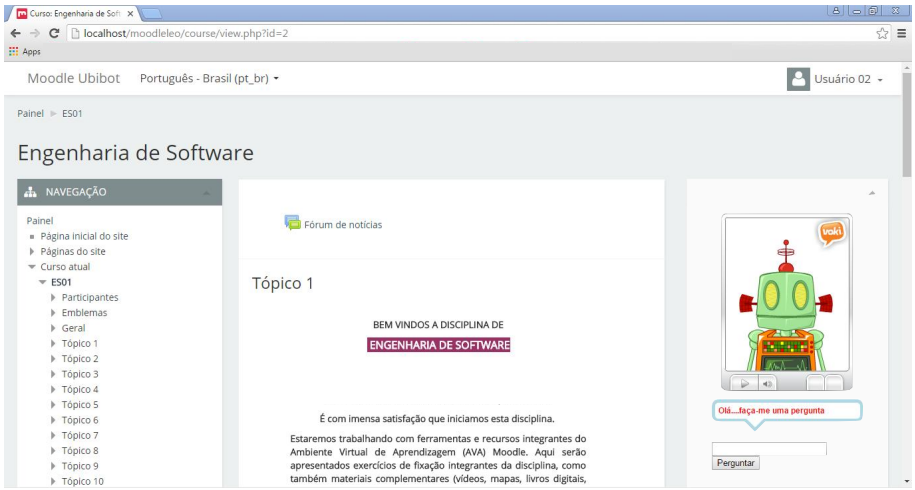
Rosa *et al.* (2017) desenvolveu um agente conversacional que visa apoiar a aprendizagem de teste de software em um mundo virtual. Ele foi criado como complemento para uma peça teatral 'Testar ou não tesar - Eis a questão', que retrata um ambiente de uma fictícia empresa de teste de software, expressa no livro intitulado 'Teste de Software' de Rios e Moreira (2013). O agente assume o papel de um especialista em teste de software, representado por um NPC. Ele é constituído por três bases de conhecimento em AIML, duas bases de interação e uma base sobre o domínio. A base sobre o domínio contém pares conversacionais sobre informações do mundo virtual em que ele está inserido. Uma base de interação contém pares conversacionais com cumprimentos e frases motivacionais.

[5] Mais informações disponíveis em: <https://www.pandorabots.com/>

A outra base contém informações sobre o conteúdo de teste de software, em que foram elencados alguns conceitos definidos por Sommerville (2016), Pressman e Maxim (2016), Rios e Moreira (2013).

O agente conversacional Ubibot foi construído para apoiar o ensino da disciplina de Engenharia de Software, focando em aspectos teóricos de Modelos de Processos de Software e Engenharia de Requisitos. Desenvolvido por Paschoal, Oliveira e Chicon (2018) na linguagem AIML, possui módulos de identificação e modificação de contexto (nível de conhecimento e nível de desempenho) do aluno escritos na linguagem PHP e utiliza banco de dados MySQL[6] para armazenar as informações contextuais sobre os estudantes. Ele possui bases de conhecimento para contemplar estudantes com conhecimento básicos, intermediários ou avançados sobre o conteúdo e bases para contemplar o desempenho dos estudantes durante a realização da disciplina. A Figura 6 ilustra uma interação com o Ubibot.

Figura 6 – Interface principal do agente Ubibot



Fonte: Paschoal, Oliveira e Chicon (2018).

Os estudos apresentados no decorrer desta seção descrevem sobre mecanismos de apoio que conseguem interagir em linguagem natural, dando dicas sobre o conteúdo, orientando o aprendiz, solucionado suas dúvidas. Durante a descrição sobre os agentes conversacionais, foi mencionado o uso de linguagens para representar o conhecimento do agente (*e.g.*, AIML). Isso ocorre porque os agentes conversacionais, em geral, foram previamente programados para conversar sobre um determinado assunto. A interação é possível, pois são constituídos de bases de conhecimento sobre assuntos. Assim, ao serem questionados por um ser humano, identificam a frase, as comparam com o que possuem em suas bases de conhecimento e emitem respostas considerando o que está previamente implementado em suas bases de conhecimento. Para identificar a pergunta

[6] Mais informações disponíveis em: <https://www.mysql.com/>.

transmitida pelo usuário, buscar em uma base de conhecimentos possíveis respostas para a pergunta, e retornar a resposta mais adequada, eles utilizam diferentes tecnologias. Algumas tecnologias usados para o desenvolvimento serão discutidas na próxima seção.

2.4.2 Tecnologias para o Desenvolvimento de Agentes

Com o intuito de revelar as diferentes tecnologias utilizadas no desenvolvimento de agentes conversacionais, Bradeško e Mladenić (2012) e Abdul-Kader e Woods (2015) analisam os agentes que foram premiados nas edições do Prêmio Loebner que compreendem o período de 1991 a 2014, identificando as abordagens técnicas e algoritmos usados. A partir da análise, identificam seis tipos de tecnologias que predominam, são elas: Análise Gramatical, Casamento de Padrões, ChatScript, Linguagem AIML, Modelos de Cadeias de Markov e Ontologias. Em ambos os estudos, a linguagem AIML se destacou em relação as demais tecnologias. No decorrer desta seção, essas tecnologias serão apresentadas.

Análise Gramatical: A tecnologia conhecida como análise gramatical (do inglês: *parsing*) foi utilizada na construção dos agentes conversacionais desenvolvidos por Weintraub (1986). Essa tecnologia consiste na aplicação de análise léxica e sintática no texto de entrada emitido/enviado pelo usuário. Durante a análise léxica, cada carácter de entrada é lido e agrupado em palavras, que são classificadas em categorias (*e.g.* palavras reservadas, símbolos especiais, dentre outros). Os caracteres lidos são agrupados para formar as unidades léxicas.

Na análise sintática, as unidades léxicas são agrupadas em estruturas sintáticas, utilizando uma série de regras de sintaxe, que fazem parte da gramática da linguagem natural. Nessa etapa, os erros sintáticos presentes no texto de entrada são reconhecidos. Portanto, a análise sintática é responsável por verificar quando uma sentença faz parte da gramática da linguagem. De acordo com Abdul-Kader e Woods (2015), os agentes conversacionais construídos com análise gramatical também utilizam funções específicas para o processamento de linguagem natural. Na análise feita por Bradeško e Mladenić (2012) é possível perceber que essa tecnologia é complementada com outras tecnologias de conversação como Casamento de Padrões.

Casamento de Padrões: O casamento de padrões (do inglês: *Pattern Matching*) é uma tecnologia que é comumente usada na implementação de muitos agentes conversacionais. Existe uma variedade de algoritmos de casamento de padrões, alguns mais complexos outros com menos complexidade. Recebem esse nome porque se trata da relação entre um conjunto de palavras-chave previamente selecionadas e um grupo de respostas associadas às palavras-chave, respeitando a ordem das palavras-chave e a prioridade entre as respostas (NEVES, 2005). Esses algoritmos são similares ao utilizado no agente conversacional Eliza. A tecnologia é utilizada para instituir a lin-

guagem AIML. Portanto, agentes conversacionais desenvolvidos em AIML utilizam a técnica de casamento de padrões.

ChatScript: ChatScript é uma linguagem em *scripts* com funcionamento baseado em regras, semelhante ao AIML (ABDUL-KADER; WOODS, 2015). De acordo com (PAZ *et al.*, 2017), trata-se de uma solução *open source* alternativa para construção de agentes conversacionais, que possibilita os plurais serem convertidos em singular, os verbos às formas infinitivas, os adjetivos e advérbios às suas formas de base. Ela também oferece um conjunto de funcionalidades como conceito de variáveis, casos, e os operadores lógicos AND e OR (ABDUL-KADER; WOODS, 2015). McNeal e Newyear (2013) mencionam que a linguagem é mais complexa que AIML e a inserção de agentes conversacionais construídos nessa linguagem em páginas *web* não é trivial. Apesar de ChatScript possuir uma maior robustez, a AIML tem como vantagem a facilidade de uso, fato esse que faz com que a linguagem AIML continua sendo adotada como a solução mais propícia no desenvolvimento de agentes, especialmente para o âmbito educacional (PAZ *et al.*, 2017).

Linguagem AIML: A linguagem AIML é utilizada para representar o conhecimento de um agente conversacional, sendo derivada do XML (*eXtensible Markup Language*) (WALLACE, 2009). Ela é constituída por um conjunto de *tags*, que é utilizado para implementar a base de conhecimento do agente. As *tags* são marcadores que representam comandos que serão seguidos por um software que irá interpretar a linguagem, conhecido como interpretador ou máquina de inferência. A base de conhecimento do agente implementado em AIML é construída por pares conversacionais, formados por perguntas e respostas. Esses pares são codificados em categorias. Nesse sentido, a interação entre o ser humano e o agente conversacional é definida pela categoria, a frase recebida pelo agente é previamente definida como um padrão de entrada e a resposta que é emitida pelo agente também é previamente definida como um modelo de resposta.

A base de conhecimento deve ser implementada previamente, utilizando as *tags* mencionadas. Todas as categorias <category> devem ficar em um arquivo que inicia com o comando <aiml> e finaliza com o comando </aiml>. A linguagem possui diversas *tags*, que tem capacidade de criar diálogos bem complexos. Ao se fazer uso de variáveis de controle, por exemplo, é possível oferecer tratamento de informação em quatro níveis linguísticos: morfossintático, semântico, discursivo e pragmático (BARROS; TEDESCO, 2016).

O reconhecimento da linguagem AIML é feito por sistemas de software, conhecidos como interpretadores (WALLACE, 2009). Os interpretadores procuram fazer o casamento de padrões buscando palavra por palavra ao invés de categoria por categoria. Para tanto, eles utilizam um algoritmo denominado Graphmaster que grava os pa-

drões de entrada informados pelo usuário em uma estrutura de árvore (WALLACE, 2009). Para localizar o padrão de resposta mais adequado para a frase de entrada, o Graphmaster faz uma busca nos padrões de entrada das bases de conhecimento até identificar uma correspondência ao que foi enviado pelo usuário. Em síntese, o algoritmo funciona da seguinte forma: identifica a seção que corresponde à primeira letra ou termo informado pelo usuário, escolhe a próxima seção que possui o início de uma letra ou sentença que corresponde à frase de entrada, repete o processo até localizar a frase correspondente à entrada.

Atualmente, existem diversos interpretadores de AIML, desenvolvidos para apoiar a implementação de agentes conversacionais nas mais diversificadas linguagens. Agentes conversacionais desenvolvidos para páginas *web* escritas na linguagem PHP, como Cocorote e Ubibot, podem utilizar o interpretador Program E ou Program O. Agentes conversacionais para serem executados como aplicações *stand-alone* de sistemas operacionais podem utilizar interpretadores em linguagem Java (*e.g.*, Program D), C/C++ (*e.g.*, Program C), dentre outras.

Modelos de Cadeias de Markov: Os modelos de cadeias de Markov (do inglês: *Markov Chain Models*) são utilizados em agentes conversacionais com o intuito de desenvolver respostas que são mais aplicáveis probabilisticamente e consequentemente mais corretas (ABDUL-KADER; WOODS, 2015). Assim, cada ocorrência de uma letra ou palavra em cada conjunto de dados textuais ocorre com uma probabilidade fixa (BRADEšKO; MLADENIć, 2012). Em alguns casos, esse modelo pode equilibrar um contra senso entre sentenças. O agente conversacional Jabberwock, vencedor do prêmio Loebner de 2003, utilizava cadeias de Markov em conjunto com a técnica de casamento de padrões, análise gramatical e gramática livre de contexto.

Ontologias: Um ontologia pode ser entendida como um conjunto de conceitos fundamentais e suas relações, que detém como os seres humanos compreendem ou interpretam o domínio em questão e permite a representação de tal compreensão de maneira formal, compreensível tanto por humanos quanto por computadores (ISOTANI; BITTENCOURT, 2015; MIZOGUCHI, 2004). É capaz de oferecer uma estrutura conceitual sobre a qual se pode desenvolver bases de conhecimento compartilháveis e reutilizáveis (ISOTANI; BITTENCOURT, 2015). Pode oferecer apoio a resolução de problemas que cercam a construção de tecnologias que utilizam bases de conhecimento (MIZOGUCHI, 2004), como os agentes conversacionais.

De acordo com Oliveira *et al.* (2011), uma ontologia permite que sejam desenvolvidas bases de conhecimento com uma riqueza semântica maior do que utilizando apenas as categorias e tópicos da linguagem AIML. De acordo com Abdul-Kader e Woods (2015) e Bradeško e Mladenić (2012) ontologias são utilizadas em um agente conversacional para representar o conhecimento do mesmo, hierarquizando e relaci-

onando os conceitos interligados, visto que uma ontologia é capaz de representar e expressar o significado de conceitos (MIZOGUCHI, 2004).

Apesar de existirem diferentes tecnologias, ainda não existe um panorama que analisa com precisão as tecnologias usadas no desenvolvimento dos agentes conversacionais já estabelecidos. Também, não há uma relação que confirme que a Linguagem AIML é a tecnologia mais adequada para representar o conhecimento. Por esse motivo, muitas vezes, os autores desenvolvem os agentes conversacionais com base nos seus conhecimentos prévios.

2.5 Considerações Finais

O teste de software é uma atividade realizada seguindo uma série de critérios e técnicas que objetivam a realização sistemática do teste. Apesar do conhecimento da área estar consolidado, nem todo o profissional da indústria do software possui tais conhecimentos, uma vez que esse profissionais são, em sua maioria, oriundos do meio acadêmico. Sabe-se, no entanto, que há um déficit na formação de alunos de computação. Assim, a área de ensino e aprendizagem de teste de software vem recebendo investigação.

Algumas iniciativas que apoiam o ensino de teste de software vêm emergindo, em especial referentes ao uso de metodologias diferentes do ensino tradicional, que ofereçam condições para o estudante protagonizar o seu processo de aprendizado. Dentre os modelos pedagógicos existentes, o *flipped classroom* surge com uma proposta interessante. No entanto, antes de adotá-lo, conforme mencionado no Capítulo 1, é necessário conduzir experimentos, com o propósito de reconhecer a viabilidade do modelo no domínio. Em razão disso, com o propósito de compreender a viabilidade desse modelo no ensino de teste de software, o Capítulo 3 apresenta experimentos que foram realizados.

Ainda, vale destacar que o modelo pedagógico *flipped classroom*, por ser relativamente novo em muitas áreas de ensino, vêm apresentando algumas limitações. Uma delas está associada ao fato dos alunos não ter suporte do professor para a resolução de dúvidas no momento "antes da aula". Diante disso, surgem os agentes conversacionais, como uma proposta de mecanismo de apoio ao modelo. Os agentes conversacionais são adotados em diferentes áreas, em especial na computação, como mecanismos que podem solucionar as dúvidas dos alunos, motivá-los, dentre outros.

CAPÍTULO

3

ESTUDOS EXPERIMENTAIS

3.1 Considerações Iniciais

O estudo sobre a viabilidade do modelo vem sendo realizado em diferentes áreas, especialmente na Computação (GIANNAKOS; KROGSTIE; CHRISOCHOIDES, 2014), tais como Introdução à Programação (AMRESH; CARBERRY; FEMIANI, 2013), Álgebra (LOVE *et al.*, 2014), dentre outras. Isso é importante porque é necessário compreender melhor as possibilidades (vantagens e desvantagens) do modelo antes de adotá-lo, ao passo que o modelo pode ser mais adequado para determinados temas.

No ensino de programação, por exemplo, Amresh, Carberry e Femiani (2013) analisaram a viabilidade do modelo, por meio da eficácia em ensinar programação aos alunos. Na sequência, o estudo de Maher *et al.* (2015) apresentou um conjunto de estratégias flexíveis para ensinar programação por meio do modelo *flipped classroom*. Diante isso, nota-se a necessidade de investigar o modelo como uma estratégia para ensinar teste de software.

Conforme mencionado no Capítulo 1, uma forma de analisar a viabilidade é por meio de estudos experimentais. Diante disso, este capítulo visa contribuir com o ensino de teste de software por meio da condução de estudos experimentais. Em especial, busca-se oferecer resultados e discussões para responder a primeira questão de pesquisa que é discutida nesta dissertação: *É viável ensinar conteúdos de teste de software com o modelo pedagógico flipped classroom?*

Três estudos experimentais foram planejados e executados, cada um deles com objetivos particulares, grupos específicos de alunos, em que diferentes variáveis que permeiam o processo de ensino de teste são colocadas sob investigação. Para apoiar a condução de cada um desses experimentos, o processo de Wohlin *et al.* (2012) é seguido. Assim, no decorrer deste capítulo, serão apresentados os três estudos experimentais, englobando em cada um deles a definição do estudo, o planejamento, as operações realizadas, a análise

dos resultados e, por fim, é apresentado uma discussão sobre o empacotamento dos dados, formulários e materiais usados.

3.2 Primeiro Experimento

Esta seção apresenta um estudo experimental preliminar com o propósito de comparar o modelo *flipped classroom* com o modelo tradicional de ensino, durante o ensino de teste de software. Para comparar os modelos foi considerado o desempenho acadêmico dos estudantes em avaliações (testes de conhecimento), como estratégia para determinar se a aprendizagem por intermédio dos modelos foi efetiva. No estudo de Amresh, Carberry e Femiani (2013), similarmente, os autores utilizaram o desempenho dos estudantes para verificar a eficácia do modelo com estudantes da disciplina de Introdução à Programação.

Adicionalmente, para complementar a comparação dos modelos, a motivação do aluno é investigada, uma vez que há estudos sobre ensino de teste de software que argumentam que os alunos não ficam motivados em aprender o conteúdo (ALLISON; JOO, 2015; PASCHOAL; SOUZA, 2018b) e os professores estão encontrando alunos pouco motivados em aprender teste de software (ALLISON; JOO, 2015). Os alunos de computação preferem programação em oposição ao aprender a realizar a atividade de teste (LEMOS *et al.*, 2015; LEMOS *et al.*, 2018).

Assim, esta seção é destinada a apresentação de todas as etapas do estudo experimental conduzido visando analisar o desempenho e a motivação dos alunos ao aprender por intermédio do modelo *flipped classroom*. Este experimento foi planejado e conduzido no período de Maio e Junho de 2017 e contou com a participação de alunos do curso de bacharelado em ciências de computação da Universidade de São Paulo, matriculados na disciplina de engenharia de software.

3.2.1 Definição do Experimento

Objetivo

O experimento deve avaliar se o modelo pedagógico *flipped classroom* (tratamento) resulta em um maior desempenho acadêmico nos alunos, e se resulta em uma maior motivação (resultado) em relação ao modelo tradicional (tratamento). Essa análise de causa (tratamento) e efeito (resultado) é baseada em duas questões de pesquisa:

Primeira Questão de Pesquisa: O modelo *flipped classroom* aumenta a motivação do aluno durante o aprendizado de teste de software?

Segunda Questão de Pesquisa: O modelo *flipped classroom* aumenta o desempenho do aluno em avaliações durante o aprendizado de teste de software?

A partir da definição das questões de pesquisa, o objetivo do experimento foi estabelecido: **Analisar** o modelo pedagógico *flipped classroom*, **com o propósito de** verificar seu impacto no ensino de teste de software, **com respeito a** motivação e ao desempenho dos alunos, **do ponto de vista** dos pesquisadores, **no contexto de** estudantes de engenheira de software, estudando o conteúdo de teste de software.

Objeto

O objeto de estudo deste experimento é uma implementação do modelo *flipped classroom* no ensino de teste de software. Essa implementação segue os direcionamentos de Bergmann e Sams (2012).

Foco Qualitativo

O foco qualitativo deste experimento é medir o desempenho obtido por alunos de engenharia de software ao estudarem teste de software com o modelo pedagógico *flipped classroom*. Também tem como foco qualitativo mensurar a motivação dos alunos perante as condições mencionadas.

Perspectiva

Considerando um estudo experimental, o ideal é conduzi-lo em um ambiente mais próximo ao real do objeto sob investigação, em que o pesquisador possa controlar os diferentes fatores envolvidos. Assim, considerando que o objeto sob investigação é o modelo *flipped classroom* no ensino de teste de software, o ambiente ideal para realização do estudo é o meio acadêmico. Diante disso, este é avaliado em um ambiente acadêmico.

Contexto

Este estudo experimental foi conduzido no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo. Ele foi realizado no contexto da disciplina de engenharia de software[1], oferecida aos alunos do 5º semestre do bacharelado em ciências de computação, durante o ensino da unidade de teste de software.

Na ocasião, os alunos da disciplina iriam começar a estudar a técnica de teste funcional. Diante disso, o estudo é conduzido no contexto de aprendizagem de teste de software com alunos de graduação em Computação, durante o aprendizado da técnica de teste funcional.

[1] Mais informações sobre a disciplina estão disponíveis em: <http://bit.ly/2FAEY7p>.

3.2.2 Planejamento

A fase de planejamento de um estudo experimental tem o propósito de estabelecer um documento de planejamento, que oferecerá suporte para a execução e análise (WOHLIN *et al.*, 2012). Essa etapa envolve a descrição de um conjunto de atividades que representam como o estudo experimental deve ser idealizado, a saber: seleção de contexto, formulação de hipóteses, seleção dos participantes, seleção das variáveis e descrição da instrumentação (TRAVASSOS; GUROV; AMARAL, 2002). Cada atividade envolvida no planejamento será apresentada durante esta seção.

Seleção do Contexto

O contexto pode ser caracterizado conforme quatro dimensões (TRAVASSOS; GUROV; AMARAL, 2002): o processo (*online*/*offline*), os participantes (alunos/profissionais), a realidade (o problema real/modelado) e a generalidade (específico/geral). Nessa perspectiva, esse experimento supõe um processo *offline* em que os alunos não são entrevistados durante toda a aplicação do modelo pedagógico, mas em momentos específicos.

Os participantes são os alunos da disciplina de engenharia de software, que estão estudando o conteúdo de teste de software. Essa estrutura de participantes é bastante comum em cursos de graduação em computação no Brasil, pois na maioria das instituições de ensino superior do Brasil a atividade de teste de software é ensinada como uma unidade da disciplina de engenharia de software (PASCHOAL; SOUZA, 2018b).

O experimento pode ser considerado real, uma vez que o impacto do modelo no ensino de teste de software é caracterizado por meio da resolução de atividades comumente realizadas no ensino de teste de software. O impacto do modelo no aprendizado dos alunos de teste de software é comparado com o impacto do modelo tradicional, então, o contexto possui um caráter específico.

Definição das Hipóteses

Com o intuito de responder as questões de pesquisa descritas na Seção 3.2.1, foram formuladas questões de pesquisa. Para a primeira questão de pesquisa ("O modelo *flipped classroom* aumenta a motivação do aluno durante o aprendizado de teste de software?"), as seguintes hipóteses foram estabelecidas:

- **Hipótese Nula:** Não existe diferença significativa em termos de motivação entre alunos que estudam com o modelo de *flipped classroom* e com o modelo tradicional.

  H_0: Motivação $_{\text{Flipped Classroom}}$ = Motivação $_{\text{Tradicional}}$

- **Hipótese Alternativa:** Existe diferença significativa em termos de motivação entre alunos que estudam com o modelo de *flipped classroom* e com o modelo tradicional.

H_1: Motivação $_{\text{Flipped Classroom}} \neq$Motivação $_{\text{Tradicional}}$

Para a segunda questão de pesquisa ("O modelo *flipped classroom* aumenta o desempenho do aluno em avaliações durante o aprendizado de teste de software?"), as seguintes hipóteses foram estabelecidas:

- **Hipótese Nula:** Não existe diferença significativa em termos de desempenho entre alunos que estudam com o modelo de *flipped classroom* e com o modelo tradicional.

  H_0: Desempenho acadêmico $_{\text{Flipped Classroom}}$ = Desempenho acadêmico $_{\text{Tradicional}}$

- **Hipótese Alternativa:** Existe diferença significativa em termos de desempenho entre alunos que estudam com o modelo de *flipped classroom* e com o modelo tradicional.

  H_1: Desempenho acadêmico $_{\text{Flipped Classroom}} \neq$Desempenho acadêmico $_{\text{Tradicional}}$

Seleção dos Sujeitos

Apesar do Instituto de Ciências Matemáticas e de Computação oferecer uma disciplina específica de teste de software aos alunos do 5º ano dos cursos de Computação, no semestre em que esta pesquisa foi realizada a disciplina possuía um tamanho reduzido de alunos. Diante disso, foi necessário recorrer à disciplina de engenharia de software oferecida pelo instituto aos alunos do 3º ano. Assim, a escolha da amostra de sujeitos para o estudo experimental foi baseada em conveniência, os estudantes mais próximos do pesquisador e mais convenientes foram selecionadas para participar do estudo.

Foram convidados para participar do experimento 106 estudantes de graduação, todos devidamente matriculados na disciplina de engenharia de software, sem conhecimento prévio sobre a técnica de teste funcional. A falta de conhecimento sobre a técnica é um requisito importante para a seleção dos sujeitos, pois neste estudo experimental os sujeitos foram expostos aos modelos pedagógicos com o propósito de aprender o conteúdo por meio dos modelos.

Os 106 estudantes foram divididos em dois grupos, de maneira randômica. Essa divisão é importante para balancear os grupos que receberão os tratamentos, facilitando posteriormente a análise estatística dos dados. Desse modo, inicialmente, foi planejado que 53 participantes fossem expostos ao aprendizado de teste funcional por meio do modelo *flipped classroom* enquanto os outros 53 participantes fossem expostos ao modelo tradicional.

É importante salientar que apesar do experimento ser realizado no contexto da disciplina de engenharia de software e envolver os conteúdos de uma unidade da disciplina, a participação dos estudantes foi voluntária e não contabilizou como atividade avaliativa da disciplina, isto é, os estudantes que participarem do experimento não foram beneficiados.

Seleção das Variáveis

Em estudos experimentais, a relação entre causa e efeito é mensurada por intermédio de variáveis. Nesse sentido, é necessário analisar os efeitos promovidos por um determinado tratamento em uma determinada variável ao final do processo. Nesse tipo de estudo existem dois tipos de variáveis: variáveis independentes e variáveis dependentes Travassos, Gurov e Amaral (2002).

As variáveis independentes, também conhecidas como variáveis de entrada, podem ser controladas pelo investigador. São elas que provocarão algum tipo de efeito nas variáveis dependentes. Já as variáveis dependentes, também conhecidas como variáveis de saída, são aquelas que são afetadas pelas variáveis independentes. Dessa maneira, o valor de uma variável dependente representa o resultado do experimento.

Variáveis Independentes

Neste experimento, as seguintes variáveis independentes foram consideradas:

- **Experiência dos participantes:** Refere-se ao nível de conhecimento dos estudantes sobre o conteúdo abordado no experimento. No contexto desse experimento, alunos de graduação que não tinham tido contato preliminarmente com a área de teste de software foram apreciados.
- **Estratégia utilizada para ensinar teste de software (fator):** Consiste nas ações desenvolvidas que representam e explicam a forma de como se aborda o conteúdo e que se concretizam por meio de práticas e interações entre professor-aluno. Este experimento recebe dois tratamentos:

  Tratamento A: O ensino ocorre por meio do modelo *flipped classroom*.

  Tratamento B: O ensino ocorre por meio do modelo tradicional.

Variáveis Dependentes

As seguintes variáveis dependentes foram consideradas:

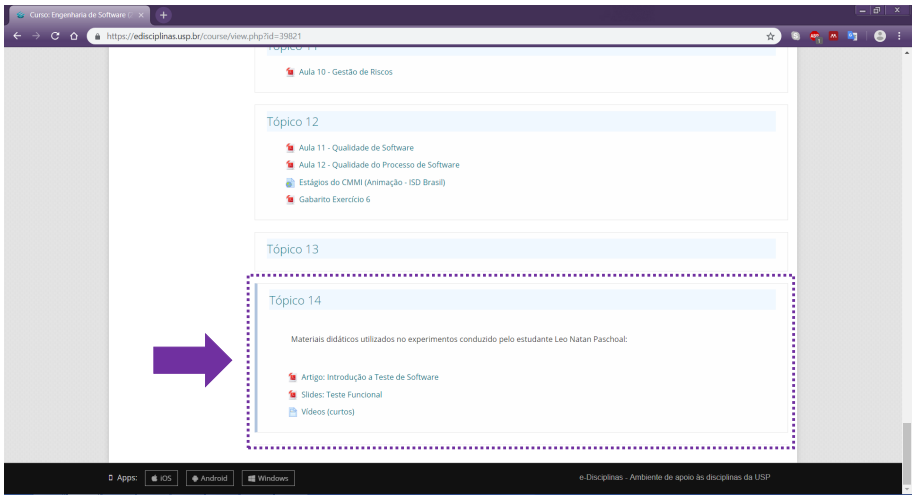
- **Motivação:** representa o grau de motivação do aluno, isto é, um resultado que indica o quão motivado o aluno ficou. Essa variável é medida por meio de um questionário específico para analisar a motivação de indivíduos.
- **Desempenho acadêmico:** o desempenho acadêmico representa o resultado obtido pelo participante em uma avaliação (teste de conhecimento). Essa variável é mensurada pela nota obtida pelo aluno em um teste de conhecimento. A nota varia em uma escala de $0 - 10$ pontos.

Descrição da Instrumentação

Durante todo o estudo, foram pré-definidos e utilizados diferentes tipos de instrumentos. Alguns deles foram empregados como apoio à implantação do modelo *flipped classroom*, outro para mensurar o desempenho acadêmico e um dele foi usado para medir a motivação dos alunos.

Para apoiar a implantação do modelo *flipped classroom* foi utilizado uma versão institucional do ambiente virtual de aprendizagem Moodle[2]. Nesse ambiente foram disponibilizados os materiais didáticos que deveriam ser acessados pelos alunos durante o momento "antes da aula". Esse ambiente foi utilizado apenas por alunos que foram instruídos por intermédio de *flipped classroom*. A Figura 7 demonstra um bloco do ambiente que foi preparado para apoiar a utilização de *flipped classroom*.

Figura 7 – Materiais didáticos disponíveis no ambiente Moodle



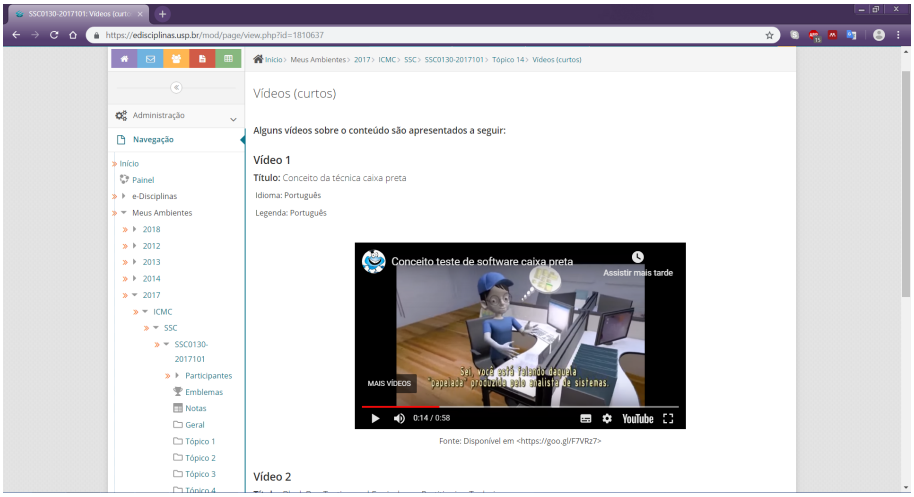
Fonte: Elaborada pelo autor.

Foi necessário pré-definir alguns recursos educacionais para apoiar o aprendizado dos alunos no momento "antes da aula". Dentre os recursos, optou-se por definir materiais didáticos em diferentes formatos, visando contemplar alunos com diferentes estilos e preferências. Assim, foram definidos recursos nos seguintes formatos: vídeos, slides e artigo.

O autor da pesquisa selecionou um conjunto de materiais e todos eles foram analisados previamente por dois especialistas na área teste de software, com a titulação de doutor há mais de 15 anos e com experiência em ministrar os conteúdos abordados. Diante disso, os materiais instrucionais utilizados foram produzidos por terceiros. Um exemplo de material didático é apresentado na Figura 8.

[2] Mais informações disponíveis em: <https://moodle.org/>

Figura 8 – Exemplo de um vídeo utilizado para instruir os estudantes



Fonte: Elaborada pelo autor.

Salienta-se que os slides utilizados pelos alunos que estudaram por meio do modelo *flipped classroom* também foram empregados durante o ensino de teste para o grupo controle. Neste caso, o professor que ministrou as aulas aproveitou esse material. O professor em questão é doutor, professor associado do instituto onde a pesquisa é realizada, com experiência há mais de 15 anos na área, ministrando a disciplina de engenharia de software e teste de software, com pesquisa sobre ensino de teste de software.

Também foram estabelecidos os materiais usados para coletar dados dos participantes. Um deles é o questionário sociodemográfico que buscou reconhecer informações sobre o gênero dos indivíduos, idade, formação e se os estudantes tinham algum tipo de experiência ou contato prévio com teste de software. Essas informações têm a finalidade de caracterizar a amostragem utilizada.

Para mensurar o desempenho dos alunos, foi preparada uma avaliação. Essa avaliação foi definida na forma de questionário composto por oito questões de múltipla escolha, que englobam aspectos teóricos sobre teste funcional. Cada uma delas possui cinco opções de resposta e apenas uma delas é correta. As questões foram selecionadas e retiradas de concurso e exames de certificação da área de teste de software. As respostas são transformadas em uma nota. Para a montagem deste questionário, os especialistas foram consultados.

A motivação dos estudantes em aprender teste de software por meio dos diferentes modelos foi mensurada por meio de um instrumento denominado IMMS (*Instructional Materials Motivation Survey*) (KELLER, 2009). Para definir o instrumento, foram analisados estudos experimentais sobre o modelo *flipped classroom* em diferentes áreas de ensino que também consideram a motivação como variável dependente. Nos estudos de Tao, Huang e Tsai (2016) e Kostaris *et al.* (2017) a motivação é considerada e mensurada

por intermédio do instrumento IMMS definido por Keller (2009).

O instrumento IMMS é constituído por 36 perguntas e cada uma delas possui cinco opções de resposta, no formato de escala Likert, que varia entre 1 até 5, sendo: (1) não é verdadeiro; (2) ligeiramente verdadeiro; (3) moderadamente verdadeiro; (4) na maior parte verdadeiro; (5) muito verdadeiro. Isso significa que a pontuação mínima na pesquisa é 36 pontos, e a máxima é 180 pontos, com uma pontuação média de 108.

As 36 questões do modelo conseguem mensurar as seguintes subescalas da motivação: atenção, relevância, confiança e satisfação. Doze questões do instrumento mensuram a atenção, nove medem a relevância, nove determinam a confiança e seis estimam a satisfação do usuário. De acordo com Keller (2009) as pontuações são determinadas pela soma das respostas de cada subescala e da escala total.

Entre as questões do IMMS há algumas que possuem afirmações negativas e, por conta disso, precisam receber pontuações invertidas durante a análise. Assim, se o aluno dar um ponto para uma questão invertida, durante a análise será necessário transformar esse valor em cinco pontos. Keller (2009) recomendam a seguinte transformação para as questões invertidas: 1 = 5, 2 = 4, 3 = 3, 4 = 2, 5 = 1.

Além dos instrumentos mencionados, um termo de consentimento foi desenvolvido, com o propósito de informar e esclarecer os estudantes selecionados sobre o experimento. A partir desse instrumento os estudantes poderiam tomar sua decisão de forma justa e sem constrangimentos sobre a sua participação no experimento.

3.2.3 Operação

A etapa de operação de um estudo experimental constitui-se pela preparação dos participantes, do material utilizado no experimento, no planejamento de implantação dos modelos e organização do ambiente em que o experimento será conduzido. Também consiste na execução do experimento, em conformidade com o planejamento pré-definido.

Preparação

Na sequência da definição e do planejamento do experimento, o professor que ministra a disciplina de engenharia de software foi contatado. Na ocasião, a professora orientadora da presente pesquisa e o pesquisador propuseram a condução do experimento. Foram discutidos assuntos como o número necessário de aulas para a condução do experimento e a necessidade de ajustes do cronograma previstos pelo professor para a condução do experimento.

Durante a preparação dos materiais, foram reunidos alguns materiais didáticos produzidos por professores de teste de software, com o propósito de oferecê-los como mecanismos de apoio ao aprendizado. Na sequência, todos os materiais didáticos foram

analisados por dois professores especialistas da área. Após a definição dos materiais didáticos, o ambiente Moodle foi preparado para disponibilizá-los. O conteúdo foi configurado no ambiente para ser visível apenas ao grupo de alunos selecionados para estudar por meio do modelo *flipped classroom*.

Testes de conhecimento foram desenvolvidos pelo pesquisador, com base em questões de concursos públicos, processos seletivos e avaliações de certificação. As questões foram definidas de forma cuidadosa, visando desafiar o participante a demonstrar o conhecimento real no momento que ele está respondendo o instrumento.

Uma versão em língua portuguesa do IMMS também foi identificada e preparada para coleta de dados. O instrumento foi originalmente escrito no idioma inglês, mas já foi traduzido para diferentes línguas, como o português, adotado no Brasil.

Para a execução do experimento, os 106 alunos foram distribuídos de maneira randômica em dois grupos. Para tanto, foram preparadas duas iniciativas de como abordar os conteúdos. Em virtude disso, duas salas de aula distintas foram necessárias, uma para a turma que estudaria com o modelo tradicional e outra para a aula com *flipped classroom*.

Para o desenvolvimento de uma aula tradicional aos alunos que estudariam teste de software por meio desse modelo, a professora orientadora dessa pesquisa preparou uma aula tradicional sobre o assunto. As preleções foram planejadas para durar todo o tempo da aula, deixando a recomendação de exercícios de fixação para ser realizada depois da aula.

Para a implementação do *flipped classroom*, os materiais didáticos foram hospedados visando facilitar o acesso, uma vez que os alunos precisariam do material para estudar no momento "antes da aula". Durante a aula, os conteúdos foram discutidos e as dúvidas dos alunos foram solucionadas. Uma atividade prática também foi conduzida com os alunos, visando exercitar o conhecimento adquirido.

A partir das decisões tomadas e da confecção de todo o material necessário para a execução do experimento, o experimento estava pronto para a execução prática. Assim, a próxima etapa do experimento foi realizada. A seguir serão apresentados os detalhes da sua execução.

Execução

O experimento foi conduzido em três fases, distribuídas no período de uma semana, compreendendo duas aulas da disciplina de engenharia de software. A primeira aula ministrada em uma terça-feira e a segunda aula em uma sexta-feira. Foram utilizadas apenas duas aulas, porque a execução do experimento precisava simular um cenário bastante próximo ao de uma aula e não poderia comprometer as atividades previstas pela disciplina. Esta seção é dedicada a apresentação das atividades que foram executadas

pelos participantes em cada uma das fases.

Fase 1: Os pesquisadores apresentaram a proposta do experimento para os alunos matriculados na disciplina, acompanhado pelo professor responsável por ministrá-la. Nesta ocasião, os alunos foram notificados que a turma seria dividida em dois grupos. Assim, uma turma de alunos experimentaria uma aula com um modelo diferente de ensino e o outro grupo experimentaria uma aula tradicional com a professora orientadora da pesquisa. Eles não foram informados que estariam sendo avaliados no decorrer do experimento, visando mitigar uma possível ameaça à validade.

No primeiro contato, foi explicado aos alunos que eles poderiam optar por participar dessa iniciativa, uma vez que haveria um outro momento especial na disciplina para retomada do conteúdo. Na sequência, os alunos que optaram por não participar do experimento foram liberados da aula. Os que aceitaram participar da pesquisa, por sua vez, preencheram um termo de consentimento, autorizando a liberação de dados.

Todos os participantes que assinaram o termo, foram convidados a responder o questionário sociodemográfico e participar de um teste de conhecimento (nesse momento denominado pré-teste). Essa atividade contém questões sobre o conteúdo de teste funcional. O objetivo da aplicação foi mensurar o conhecimento prévio que os alunos possuíam sobre o conteúdo, antes de ter contato com o conteúdo na disciplina.

Na sequência, os alunos foram informados sobre a distribuição dos grupos. Os estudantes que foram selecionados para compor o grupo experimental são aqueles que participaram das aulas com o modelo *flipped classroom* e os selecionados para compor o grupo controle são aqueles que estudaram por meio do modelo tradicional de ensino.

Após a distribuição dos participantes, os integrantes do grupo controle foram liberados da aula e não receberam nenhuma orientação sobre o experimento, já os integrantes do grupo experimental receberam orientações sobre as ações que deveriam realizar. Essas orientações estão relacionadas ao estudo no momento "antes da aula", ao incentivo em anotar as dúvidas que emergem durante o momento de estudo, a participação em uma sala de aula distinta na próxima aula.

Fase 2: Os alunos que compuseram o grupo experimental estudaram o conteúdo, em ambiente externo à sala de aula. Eles poderiam escolher qualquer ambiente para realizar o estudo, podendo fazê-lo no laboratório da universidade, biblioteca, entre outros. Além disso, poderiam estudar individualmente ou em equipes.

Nessa fase da pesquisa, os estudantes designados para o grupo controle não tinham conhecimento sobre as atividades de estudo realizadas pelo grupo experimental. Os pesquisadores haviam notificado os alunos do grupo experimental que os alunos do

grupo controle não deveriam saber o que os seus colegas estariam fazendo. Adicionalmente, os alunos do grupo experimental foram informados que não poderiam passar qualquer material ao aluno do grupo controle. Essa medida foi tomada para evitar que alunos do grupo controle tivessem acesso aos materiais destinados apenas ao grupo experimental.

Fase 3: A última fase foi dividida em três etapas:

- **Primeira etapa:** Os alunos do grupo controle participaram de uma aula tradicional. Nessa ocasião, a professora ministrou o conteúdo em um período de aproximadamente 50 minutos, equivalente a uma aula na disciplina. Durante a aula, os alunos poderiam fazer questionamentos. Exemplos também foram demonstrados, mas nenhum tipo de atividade ativa foi proposta e realizada em sala de aula. Ao final da aula, os exercícios disponíveis nos slides foram disponibilizados para que os alunos pudessem realizar como tarefa para casa, "depois da aula". Esses alunos também receberam um formulário para responder. O formulário em questão trata-se do IMMS.
- **Segunda etapa:** A segunda etapa consiste em um momento "durante a aula" para o grupo de alunos que estavam estudando com o modelo *flipped classroom*. Na ocasião, os alunos foram incentivados a participar, solucionar as suas sobre o conteúdo. Também houve um espaço para realização de uma atividade em que os alunos tiveram que por em prática os conhecimentos obtidos durante o primeiro momento. A atividade deveria ser realizada individualmente por cada um dos alunos do grupo, com suporte oferecido pela professora. Ao final da interação, os alunos tiveram que responder as questões que compõe o IMMS.
- **Terceira etapa:** Após as intervenções realizadas e atividades realizadas pelos participantes dos dois grupos, os estudantes de ambos os grupos voltaram a sala de aula. Nessa etapa, eles tiveram que realizar uma avaliação (teste de conhecimento). A atividade disponibilizada para os estudantes foi utilizada nesta fase com o intuito de mensurar o conhecimento que os alunos adquiriram sobre o conteúdo, após ter contato com o conteúdo na disciplina (pós-teste).

3.2.4 Análise dos Resultados

Após a execução do experimento e coleta dos dados, a próxima etapa realizada foi a análise de dados. Esta seção apresenta os resultados obtidos por meio do experimento e as análises feitas nesses dados. Inicialmente, serão apresentados os dados sociodemográficos sobre os participantes, na sequência é descrita uma análise descritiva e, por fim, os testes de hipótese, a fim de obter as considerações sobre o estudo.

Caracterização dos participantes

Durante a realização do estudo experimental, alguns participantes optaram por não disponibilizar os seus dados para serem analisados e utilizados nesta dissertação, outros preferiram não participar do experimento. Além disso, no decorrer da condução do experimento, alguns estudantes que aceitaram participar não compareceram em todas as atividades do experimento, outros não finalizaram o questionário de motivação (*i.e.*, esqueceram de responder alguma questão do instrumento), promovendo a redução da amostragem.

Diante disso, foram considerados apenas os dados dos estudantes que participaram completamente das intervenções, aqueles que responderam todos os formulários que os pesquisadores disponibilizaram e não esqueceram de qualquer item do questionário IMMS. Por consequência dessa decisão, restaram 55 participantes. Desse número de alunos, 24 estudaram por intermédio do modelo tradicional e 31 estudaram por meio de *flipped classroom*.

Para apoiar a caracterização dos sujeitos que participaram do estudo, foram elaboradas duas perguntas. A primeira pergunta teve o propósito de verificar se eles já possuíam alguma experiência na área e/ou contato com a atividade de teste de software. A segunda questão, uma questão optativa, solicitava que os participantes descrevessem sobre suas experiências, englobando tempo de experiência na área, critérios, técnica e ferramentas utilizadas na atividade.

Na primeira questão foram oferecidas duas opções de resposta, sim e não. Já a segunda pergunta possuía um campo aberto para resposta, que possibilitava aos participantes oferecer uma responda simples e/ou completa sobre a sua experiencia. Como resultado na primeira questão, todos os 55 participantes mencionaram que não possuíam conhecimento/expertise na área de teste. Por último, nenhum participante respondeu a segunda pergunta, uma vez que ela só precisaria ser respondida quando o aluno indicasse que possuía alguma experiência na área.

Motivação

A análise da motivação é feita considerando a descrição para uso do instrumento relatada no estudo de Keller (2009). Assim, durante a análise foram consideradas as categorias que fazem parte da motivação, são elas: (i) a atenção, (ii) a relevância, (iii) a confiança e, por último, (iv) a satisfação do participante. Essas categorias, de acordo com Keller (2009), suprem muitos dos conceitos e variáveis específicas que caracterizam a motivação humana.

Em razão da aplicação do instrumento IMMS em cada grupo do experimento, foram calculados os escores de atenção, relevância, confiança e satisfação para cada sujeitos

que participou do experimento. Também foi calculado o escore geral, que indica o escore de motivação do participante. A partir do cálculo de escore por grupo formado, é possível fazer análises para compreender qual modelo pedagógico deixou os sujeitos mais motivados. Assim, no decorrer da análise, cada uma das categorias de motivação é comparada entre os grupos. Por fim, os escores gerais de motivação são analisados.

No decorrer desta seção serão apresentadas as análises dos escores envolvendo as categorias de motivação e o escore de motivação de duas maneiras distintas. A primeira forma de análise é por meio de estatística descritiva, em que são consideradas as medidas descritivas de posição, medidas de descrição e análise gráfica (*box plot*). A segunda estratégia para análise de dados é a estatística inferencial, em que são realizados testes de hipótese. Para realização dos testes estatísticos, foi adotado um intervalo de confiança para $\alpha = 0{,}05$ e o software BioEstat (versão 5.3).

Estatística Descritiva

A primeira categoria do IMMS analisada foi a atenção. Durante a análise dos escores dessa categoria, foi possível observar que o grupo experimental obteve a maior média. Isso pode indicar que o aprendizado de teste de software por meio do modelo *flipped classroom* pode ter obtido mais atenção por parte dos alunos em comparação com o modelo tradicional. É possível observar que a mediana no grupo experimental é superior a mediana do grupo controle.

Os resultados da mediana indicam que 50% dos escores do grupo experimental na categoria de atenção foram superiores a 2,75. Em contrapartida, a mediana do grupo controle indica que 50% dos escores são inferiores a 2,42. O desvio padrão amostral também indica que os escores de atenção do grupo experimental são menos dispersos que os escores do grupo controle. É importante dar atenção aos escores mínimos e máximos, uma vez que eles justificam a maior dispersão dos escores do grupo controle.

Após a análise descritiva dos dados da dimensão de atenção, foi analisada a categoria de relevância. Nessa categoria, notou-se que o grupo controle apresentou maiores escores. Apesar da média de ambos os grupos for superior a metade da pontuação máxima de escores possíveis na escala (máximo 5), o grupo experimental apresentou uma média de 2,62. Em contrapartida, o grupo controle obteve uma média de 2,79 na categoria.

O grupo controle possui resultados mais satisfatórios em termos de escore que o grupo experimental. A mediana do grupo experimental é igual a 2,11, enquanto que a mediana do grupo controle é de 2,22. O desvio padrão amostral também revela que o grupo controle possui escores menos dispersos que o grupo experimental. Esses resultados podem ter influência dos escores mínimos e máximos. O grupo controle teve como escore mínimo 1,78 e o escore máximo de 5. O grupo experimental teve como escore mínimo 1,44

e escore máximo de 4,56.

Com base nesta análise, observa-se que os estudantes de teste de software acharam mais relevante o modelo tradicional do que o modelo *flipped classroom*. De acordo com Santana, Araújo e Bittencourt (2018), a análise dos escores dessa categoria podem revelar a sensação dos alunos sobre a relevância do curso por meio do modo como ele é ensinado, isto é, se o modo como o conteúdo é ensinado parece ser relevante ou não para a carreira do estudante.

Outra categoria analisada foi a confiança. Nessa categoria é possível notar que a média e a mediana de escores do grupo experimental é superior ao grupo controle. Apesar dessas medidas descritivas demonstrarem superioridade dos escores do grupo experimental, analisando o escore máximo da amostragem por grupo, nota-se que o grupo controle teve valores maiores. A análise do escore mínimo para cada grupo, também revela que o grupo experimental teve um escore mínimo superior ao do grupo controle.

Para um melhor entendimento do por que a mediana e média do grupo experimental ser superiores ao do grupo controle, foi calculado o desvio padrão. O desvio padrão amostral revela que os escores dos participantes do grupo controle ($s = 0{,}66$) são mais dispersos que os escores do grupo experimental ($s = 0{,}48$). Diante da análise sobre a categoria, acredita-se que os estudantes do grupo experimental se sentiriam mais confiantes sobre o sucesso no aprendizado do que os estudantes do grupo controle.

A satisfação foi a última categoria analisada das dimensões que compreendem o modelo ARCS. De maneira semelhante as categorias de confiança e atenção, nota-se que o grupo experimental obteve uma média e uma mediana superior ao do grupo controle. Enquanto que o grupo controle obteve média de 2,41, o grupo experimental obteve 2,57. Em relação à mediana, o grupo controle pontuou 2,08, já o grupo experimental obteve 2,5.

Ambos os grupos tiveram 1 escore como valor mínimo, o que pode indicar que em ambos os grupos um ou mais participantes não ficaram satisfeitos com os modelos pedagógicos. Uma análise para verificar o valor máximo também foi realizada. Notou-se que o grupo controle apresentou um valor máximo maior que o grupo experimental (grupo controle = 5; grupo experimental = 4,67). Isso colabora com os dados obtidos no desvio padrão amostral, que indicam que os escores do grupo controle são mais dispersos que os escores do grupo experimental (grupo controle$_s$ = 1,09; grupo experimental$_s$ = 0,96).

Por fim, foi analisado a motivação de modo geral. Os valores das médias para o grupo controle e experimental foram próximos, 2,6 e 2,67, respectivamente. Os valores das medianas podem indicar que os estudantes do grupo experimental ficaram mais motivados que os alunos do grupo controle, isto é, o grupo experimental possui uma mediana equivale a 2,72 enquanto que a mediana do grupo controle foi 2,58.

Analisando os valores mínimos, nota-se que o menor escore obtido pelo grupo controle foi de 1,64, enquanto que o menor escore obtido pelo grupo experimental foi inferior a esse valor. Os valores máximos também indicam que o grupo controle se saiu melhor em escores, uma vez que obteve 5 pontos contra 3,78 do grupo experimental. O desvio padrão amostral mostra que os escores foram menos dispersos no grupo experimental que no grupo controle.

Nota-se por meio da estatística descritiva que os escores da motivação nos grupos controle são próximos aos do grupo experimental. Apesar dos escores do grupo experimental demonstrarem que há uma tendência para serem superiores aos escores do grupo controle, a diferença entre eles não é significativa. Todos os dados foram tabulados e estão disponíveis na Tabela 1.

Tabela 1 – Estatística descritiva da pontuação dos participantes no questionário IMMS

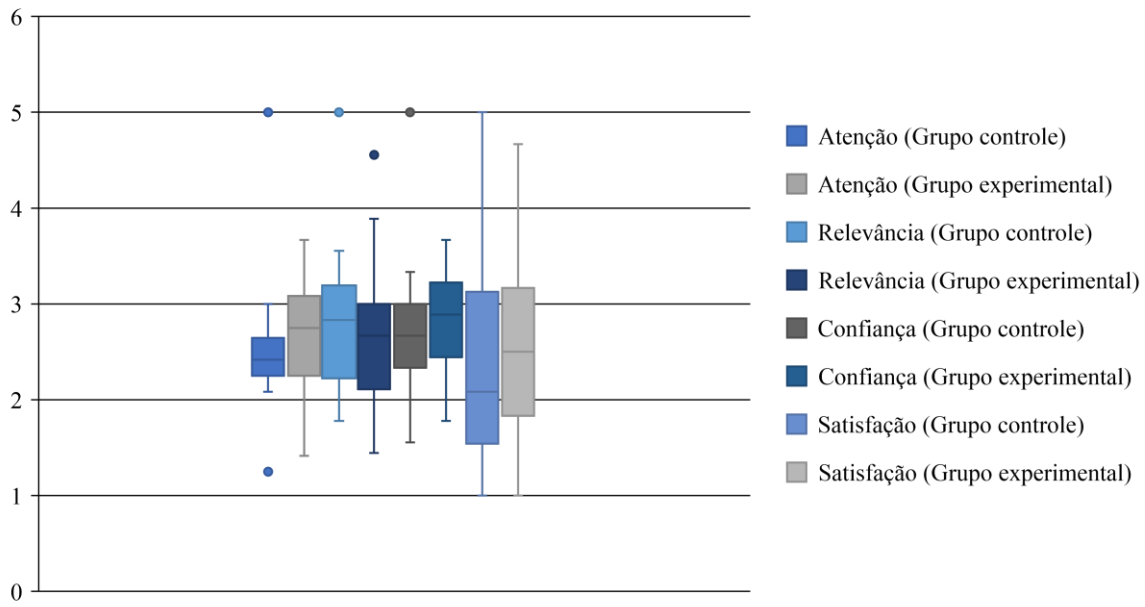
Categoria	Grupo	Média	Desvio-Padrão	Mínimo	Máximo	Mediana	Q_1	Q_3
Atenção	Controle	2,5	0,64	1,25	5	2,42	2,25	2,65
	Experimental	2,64	0,49	1,42	3,67	2,75	2,25	3,08
Relevância	Controle	2,79	0,70	1,78	5	2,83	2,22	3,19
	Experimental	2,62	0,74	1,44	4,56	2,67	2,11	3
Confiança	Controle	2,68	0,66	1,56	5	2,67	2,33	3
	Experimental	2,83	0,48	1,78	3,67	2,89	2,44	3,22
Satisfação	Controle	2,41	1,09	1	5	2,08	1,54	3,13
	Experimental	2,57	0,96	1	4,67	2,5	1,83	3,17
Geral	Controle	2,6	0,63	1,64	5	2,58	2,26	2,8
	Experimental	2,67	0,55	1,61	3,78	2,72	2,25	3,06

Fonte: Elaborada pelo autor.

Buscando representar graficamente as análises estatísticas dos dados e reconhecer possíveis *outliers*, foram gerados gráficos do tipo *box plot*. A Figura 9 busca comparar

graficamente os conjuntos de escores com respeito à media, a dispersão e distribuição em cada uma das categorias do modelo ARCS. Assim, foram criados os *box plots* para cada categoria, paralelamente, no mesmo gráfico.

Figura 9 – *Box plot* dos escores das categorias do IMMS



Fonte: Elaborada pelo autor.

Em relação as caixas geradas a partir dos escores da categoria de atenção, observa-se que a distribuição dos escores de atenção estão mais concentrados na caixa do grupo controle. A geração dos *box plots* revelou a presença de valores discrepantes (*outliers*) nos escores do grupo controle. É possível notar também que o grupo experimental obteve escores com uma alta amplitude e maior dispersão.

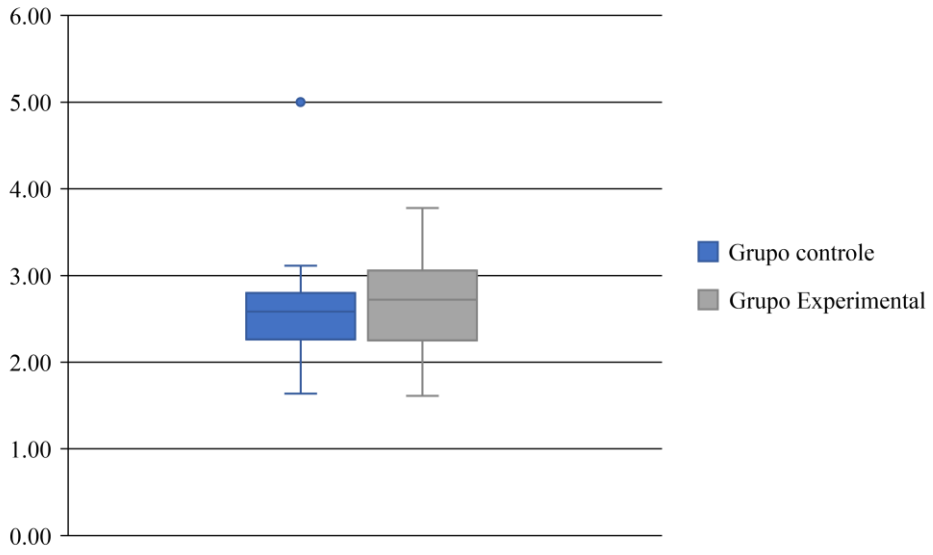
Considerando as caixas relacionadas à categoria de relevância, nota-se que os limites do grupo experimental se sobrepõem aos do grupo controle. A variabilidade dos dados são aproximadas. Os *box plots* revelam uma alta variabilidade dos escores nos dois grupos, o que torna difícil o entendimento sobre as tendências existentes nos grupos, pois alguns escores estão muito próximos ao quartil inferior e outros estão muito próximos ao quartil superior. A presença de *outliers* foi revelada, tanto no grupo controle quanto no grupo experimental.

Sobre a categoria de confiança, nota-se a existência de dados simétricos no grupo controle, indicando que nesse grupo a média é bastante próxima à mediana e à moda. Já no grupo experimental observa-se que a distribuição dos dados possui uma assimetria positiva, indicando que a maior parte dos escores dos participantes para essa escala foram baixos, enquanto poucos escores foram muito altos. A categoria também possui um *outlier* no grupo controle.

Quanto a categoria de satisfação, nota-se que em ambos os grupos as hastes estendem-se até os valores mínimo e máximo da variável sob análise. Por consequência disso, os escores individuais atípicos que distorceriam os valores de tendência central e aumentariam a dispersão dos dados não existem. Portanto, como não há dado que ultrapassa os limites das hastes, não existem *outliers* nesses grupos para esta categoria. Por fim, nota-se que a medida de tendência central do grupo experimental é superior ao do grupo controle.

Os *box plots* para os escores de motivação em cada um dos grupos também foram gerados. Conforme é possível observar na Figura 10, os escores do grupo controle são menos dispersos que os escores do grupo experimental. Isso permite concluir que os escores de motivação do grupo experimental apresentam maior variabilidade que os escores de motivação do grupo controle.

Figura 10 – Comparação entre os escores de motivação dos grupos



Fonte: Elaborada pelo autor.

A medida de tendência central do grupo experimental é superior ao do grupo controle. Essa medida se aproxima do quartil superior do grupo controle. Também se nota que no grupo experimental a distribuição dos dados possui uma assimetria negativa, indicando que a maior parte dos escores dos participantes para esse grupo foram altos, enquanto poucos escores foram muito baixos. Nessa perspectiva, há um indício que o escore de motivação cresce no grupo experimental. Ainda, o *box plot* revela a presença de um *outlier* no grupo controle.

Visando compreender como ficou a distribuição entre os grupos de estudantes mais motivados e menos motivados, os participantes foram classificados de acordo com a sua motivação. Para fazer a classificação, foram seguidos os procedimentos descritos no estudo

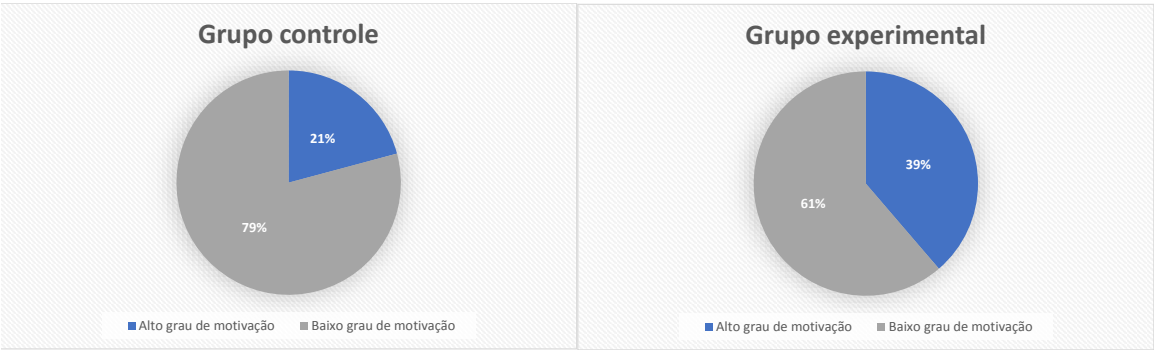
de Rondon-Melo e Andrade (2016), em que a mediana da pontuação no questionário IMMS é utilizada.

Para a classificação, as respostas dos estudantes no questionário IMMS (convertidas na escala Likert) foram somadas. Assim, os estudantes que obtiveram um somatório inferior à mediana (mediana = 105 pontos) foram classificados com baixo grau de motivação, enquanto que os sujeitos que obtiveram uma pontuação igual ou superior à mediana foram classificados com alto grau de motivação.

A Figura 11 apresenta a frequência de estudantes com baixa e alta motivação. Ao lado esquerdo, o gráfico revela que apenas 21% dos estudantes do grupo controle apresentaram uma alta motivação com o modelo pedagógico. Ao lado direito da figura, nota-se que 39% dos estudantes do grupo experimental apresentaram uma alta motivação. Nesse sentido, a porcentagem de estudantes menos motivados é maior no grupo controle em relação ao grupo experimental.

Salienta-se, entretanto, que a frequência de baixa motivação foi superior a frequência de alta motivação, independente do grupo analisado. Assim, os índices de motivação revelam que os estudantes ficaram pouco motivados aos estudar o conteúdo de teste com os modelos pedagógicos.

Figura 11 – Grau de motivação dos sujeitos, considerando o grupo controle (esquerda) e o grupo experimental (direita)



Fonte: Elaborada pelo autor.

Teste de Hipóteses

Após a análise descritiva dos dados, foram realizados testes estatísticos utilizando técnicas de estatística inferencial. Semelhante a análise descritiva, primeiramente serão apresentados os dados referentes a cada uma das categorias do IMMS. Na sequência, visando responder a primeira questão de pesquisa deste experimento e tomar uma decisão entre as duas hipóteses estabelecidas, foram feitos os testes de hipótese.

Antes da condução das análises foram consideradas as seguintes definições para aplicação de teste estatístico. Para analisar a normalidade dos dados, em todos os casos,

definiu-se o Teste de Shapiro-Wilk. Em virtude dos dados ser não paramétricos, em caso de distribuição gaussiana, foi definido que seria usado o Teste t de Student. Em caso de distribuição não gaussiana, foi definido o Teste de Mann-Whitney.

Analisando a distribuição dos escores da categoria de atenção, o valor de p para o grupo controle foi de $0{,}0082$ (isto é, $\mathsf{p\text{-}value} < \alpha$) enquanto que o valor de p para o grupo experimental foi de $0{,}7948$ (isto é, $\mathsf{p\text{-}value} > \alpha$). Por consequência de um grupo não possui distribuição normal, o Teste de Mann-Whitney foi utilizado. O valor de p revelado pelo Teste de Mann-Whitney foi de $0{,}0417$ (isto é, $\mathsf{p\text{-}value} < \alpha$). A partir da análise, é possível inferir que há uma diferença significativa entre os escores da categoria de atenção. Os dados completos estão descritos na Tabela 2.

Tabela 2 – Teste de Mann-Whitney – Categoria de Atenção

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Soma dos Postos (Ri)	570,0	970,0
Mediana	2,42	2,75
U	270,00	
Z (U)	1,7311	
p-valor (unilateral)	0,0417	
p-valor (bilateral)	0,0834	

Fonte: Elaborada pelo autor.

A distribuição dos escores referentes à categoria de relevância foram analisados. O valor de p para o grupo controle foi de $0{,}0275$ (isto é, $\mathsf{p\text{-}value} < \alpha$). Em contraponto, o valor de p para o grupo experimental foi de $0{,}3915$ (isto é, $\mathsf{p\text{-}value} > \alpha$). De maneira semelhante a análise feita nos dados da categoria de atenção, para esta categoria, o Teste de Mann-Whitney foi adotado. O valor de p obtido por meio do Teste de Mann-Whitney resultou em $0{,}1753$ (isto é, $\mathsf{p\text{-}value} > \alpha$). Com base nisso, infere-se que não existe diferença estatisticamente significativa entre os escores dessa categoria. Os dados obtidos na análise estão presentes na Tabela 3.

Tabela 3 – Teste de Mann-Whitney – Categoria de Relevância

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Soma dos Postos (Ri)	727,0	813,0
Mediana	2,83	2,67
U	317,00	
Z (U)	0,9334	
p-valor (unilateral)	0,1753	
p-valor (bilateral)	0,3506	

Fonte: Elaborada pelo autor.

Foram analisados os escores da categoria de confiança, quanto a sua distribuição. Em relação aos escores do grupo controle, obteve-se um valor de p igual a $0{,}0097$, o que

indicam uma distribuição anormal (isto é, p-value $< \alpha$). Em contrapartida o valor de p para os escores do grupo experimental foram de 0,5159, indicando uma distribuição normal (isto é, p-value $> \alpha$). Na sequência, utilizou-se o Teste de Mann-Whitney para verificar se nesta categoria os escores do grupo controle são equivalentes aos escores do grupo experimental. Obteve-se como resultado o valor de p igual a 0,0677 (isto é, p-value $> \alpha$). Assim, semelhantemente a categoria de relevância, os escores do grupo controle não diferem significativamente dos escores do grupo experimental. A Tabela 4 contém os resultados completos.

Tabela 4 – Teste de Mann-Whitney – Categoria de Confiança

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Soma dos Postos (Ri)	584,0	956,0
Mediana	2,67	2,89
U	284,00	
Z (U)	1,4935	
p-valor (unilateral)	0,0677	
p-valor (bilateral)	0,1353	

Fonte: Elaborada pelo autor.

A distribuição dos escores da última categoria do IMMS (satisfação) também foi analisada. O valor de p do grupo controle foi de 0,5590 (isto é, p-value $> \alpha$). Já o valor de p do grupo experimental resultou em 0,0903 (isto é, p-value $> \alpha$). Assim, a distribuição dos escores de ambos os grupos foi considerada normal. Por isso, o Teste t de Student para amostras independentes foi utilizado. A partir disso, observou-se o valor de p resultante foi de 0,2824 (isto é, p-value $> \alpha$), indicando que para essa categoria, não houve diferença significativa entres os escores dos grupos. A Tabela 5 contém a análise completa do teste executado.

Tabela 5 – Teste t de Student – Categoria de Satisfação

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Média	2,4097	0,9181
Variância	1,1859	0,9181
	Homocedasticidade	
Variância	1,0343	
t	-0,5792	
Graus de liberdade	53	
p-valor (unilateral)	0,2824	
p-valor (bilateral)	0,5649	

Fonte: Elaborada pelo autor.

Finalmente, a motivação foi analisada. Essa análise surgiu a partir das hipóteses estabelecidas para a primeira questão de pesquisa deste experimento, são elas: H_0: Motivação $_{\text{Flipped Classroom}}$ = Motivação $_{\text{Tradicional}}$; e, H_1: Motivação $_{\text{Flipped Classroom}}$ ≠Motivação $_{\text{Tradicional}}$.

Para fazer o teste de hipótese, a distribuição dos dados por grupo, inicialmente, foi analisada. O valor de p para o grupo controle resultado em 0,0090 (isto é, p-value $< \alpha$), indicando que os escores de motivação desse grupo não seguem uma distribuição normal. O valor de p para o grupo experimental, por sua vez, foi 0,5590 (isto é, p-value $> \alpha$), indicando que os escores de motivação desse grupo possuem uma distribuição gaussiana.

Na sequência, o Teste de Mann-Whitney foi realizado. O valor de p obtido foi igual a 0,4293 (isto é, p-value $> \alpha$), indicando que não há diferença significativa entre os escores de motivação entre os grupos. Portanto, a hipótese nula não é rejeitada. Na Tabela 6 os dados dos testes são apresentados.

Tabela 6 – Teste de Mann-Whitney - Motivação

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Soma dos Postos (Ri)	661,5	878,5
Mediana	2,58	2,5
U	361,5	
Z (U)	0,1782	
p-valor (unilateral)	0,4293	
p-valor (bilateral)	0,8586	

Fonte: Elaborada pelo autor.

Desempenho acadêmico

Esta seção tem a finalidade de apresentar os resultados relacionados ao desempenho acadêmico. Busca-se com a análise responder a seguinte questão: "O modelo *flipped classroom* aumenta o desempenho do aluno em avaliações durante aprendizado de teste de software?". Inicialmente, será apresenta uma análise dos dados, baseando-se em estatística descritiva e por último serão apresentados os testes de hipótese.

Estatística Descritiva

Para analisar o desempenho dos alunos, foram aplicados e, consequentemente, analisados duas avaliações (testes de conhecimento). Ambas foram realizadas tanto pelos alunos do grupo controle quanto pelos alunos do grupo experimental. A primeira avaliação teve apenas o objetivo de caracterizar o conhecimento inicial dos participantes sobre a área. Apesar dos estudantes mencionarem que não possuíam experiência/expertise na área, achou-se necessário investigar o conhecimento inicial.

A partir das respostas dos participantes nas oito questões que compreendem a primeira avaliação, o número de acertos foi transformado em uma escala que varia entre 0 e 10 pontos, com regra de três. A partir da transformação, os dados foram tabulados e analisados, inicialmente, por meio de estatística descritiva. A Tabela 7 contém a análise descritivas das pontuações dos participantes.

Nota-se que os estudantes do grupo experimental obtiveram uma média de pontuação superior aos estudantes do grupo controle. Apesar disso, a mediana revela que em ambos os grupos, mais de 50% das pontuações não ultrapassaram 2,5 pontos. Nessa análise, foi possível constatar que nos dois grupos houve alunos que não marcaram nenhum ponto na avaliação. A pontuação máxima do grupo experimental é superior a pontuação máxima do grupo controle. Esses valores contribuíram para aumentar a dispersão dos dados no grupo experimental, uma vez que o desvio padrão amostral é maior no grupo experimental em relação ao grupo controle.

Tabela 7 – Estatística descritiva do conhecimento prévio dos participantes

Grupo	Média	Desvio-Padrão	Mínimo	Máximo	Mediana	Q_1	Q_3
Controle	2,94	1,57	0	6,25	2,5	1,25	3,75
Experimental	3,23	1,80	0	7,5	2,5	2,5	4,69

Fonte: Elaborada pelo autor.

Após as intervenções realizadas, a avaliação final foi aplicada (pós-teste) visando analisar o desempenho dos estudantes em atividades sobre a técnica de teste funcional. De maneira semelhante à avaliação inicial, os acertos dos participantes no teste foram convertidos em uma nota. A partir da análise de notas (Tabela 8), observa-se que a média e a mediana do grupo experimental foram superiores aos valores do grupo controle.

Tabela 8 – Estatística descritiva do desempenho dos alunos

Grupo	Média	Desvio-Padrão	Mínimo	Máximo	Mediana	Q_1	Q_3
Controle	5,31	2,13	2,5	10	5	3,75	7,5
Experimental	5,93	1,68	1,25	8,75	6,25	5	7,5

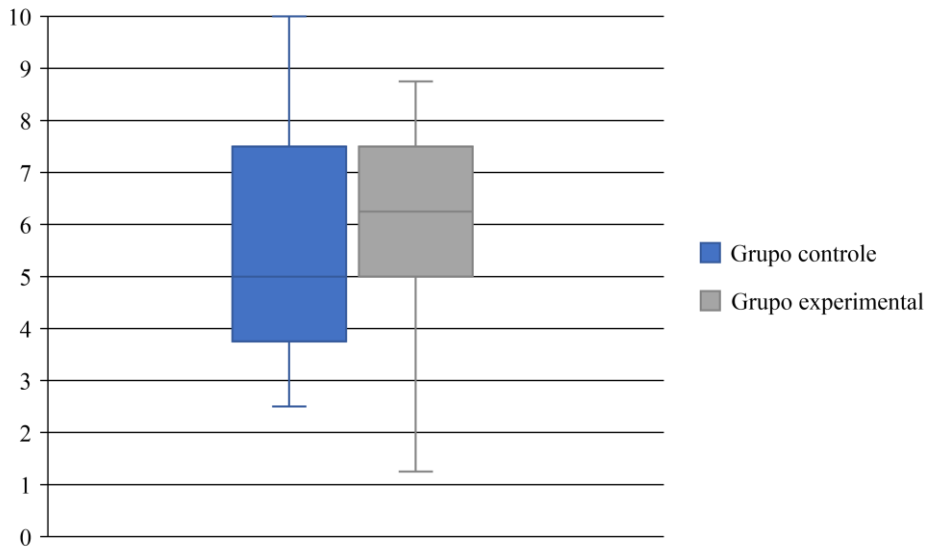
Fonte: Elaborada pelo autor.

Estudantes do grupo controle conseguiram conquistar a pontuação máxima da avaliação (10 pontos), enquanto que o grupo experimental conquistou uma pontuação máxima de 8,75 pontos. Além disso, o mínimo de ponto obtido pelos estudantes do grupo

controle foi 2,5 pontos, o que indica que eles acertaram pelo menos 25% da avaliação. Enquanto isso, o grupo experimental teve uma pontuação mínima de 1,25. Contudo, o desvio padrão amostral revela que o desempenho do grupo experimental é menos dispersa que o desempeno do grupo controle.

As notas obtidas nas avaliações, as quais representam o desempenho dos alunos, foram utilizadas para construção de *box plots*. A Figura 12 contém dois *box plots*, um para o desempenho do grupo controle e, ao lado, o outro com o desempenho do grupo experimental. Em uma análise inicial, nota-se que não houve em nenhum dos grupos a presença de *outliers*. Ao fazer uma comparação ente as medidas de tendência central, nota-se que o grupo experimental é superior ao do grupo controle.

Figura 12 – *Box plot* que compara o desempenho dos alunos



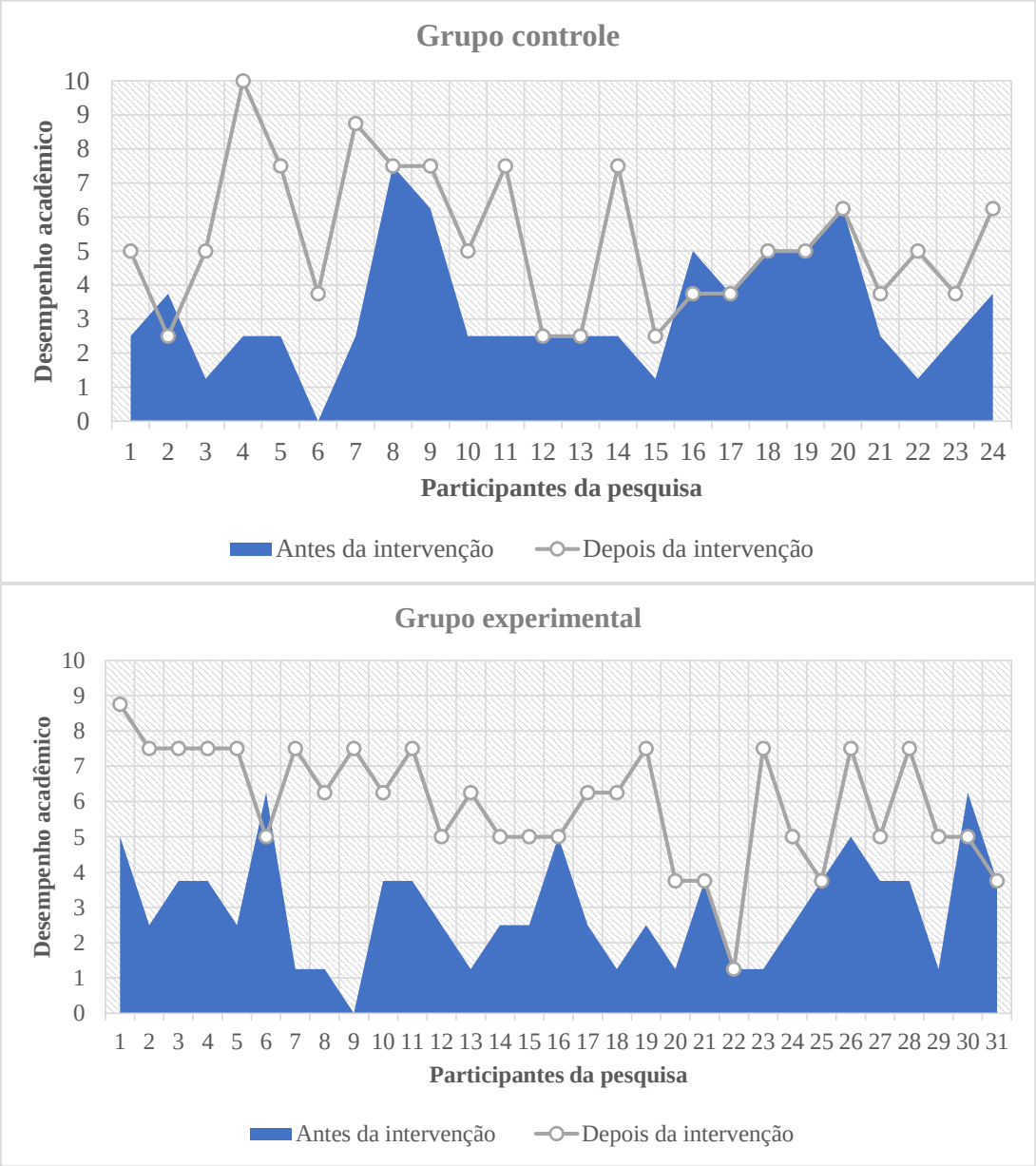
Fonte: Elaborada pelo autor.

Os desempenhos individuais dos alunos nas pontuações das atividades (antes da intervenção e depois da intervenção) foram comparadas. A Figura 13 apresenta a comparação realizada. Observa-se que o desempenho dos estudantes cresce na medida em que eles participaram da intervenção, algo que era naturalmente esperado. Apesar do desempenho crescer no grupo controle, nota-se que o crescimento não segue um padrão. Alguns alunos obtêm excelentes desempenho em relação à avaliação inicial, enquanto que outros aumentam pouco o desempenho.

Testes de Hipótese

A partir da análise descritiva nos dados de desempenho dos estudantes, foram realizadas análises com técnicas de estatística inferencial. Essas análises foram necessárias,

Figura 13 – Comparação de conhecimento prévio versus desempenho no grupo controle e no grupo experimental



Fonte: Elaborada pelo autor.

pois buscou-se testar as hipóteses definidas no início da Seção 3.2.2. Durante a definição de hipótese foram criadas duas hipóteses para a variável de desempenho, são elas: H_0: Desempenho acadêmico $_{\text{Flipped Classroom}}$ = Desempenho acadêmico $_{\text{Tradicional}}$; e, H_1: Desempenho acadêmico $_{\text{Flipped Classroom}}$ $\neq$Desempenho acadêmico $_{\text{Tradicional}}$.

Antes de realizar o teste de hipótese, foi pré-estabelecido que a distribuição dos dados seria analisada por meio do Teste de Shapiro-Wilk. Semelhantemente ao teste usado na análise de hipótese sobre a variável de motivação, foi definido que seria utilizado o Teste t de Student em caso de distribuição normal dos dados e o Teste de Mann-Whitney em

caso de distribuição anormal.

Durante a análise de normalidade, o grupo controle obteve o valor de p igual a 0,0845 (isto é, p-value $> \alpha$), indicando que o desempenho desse grupo segue uma distribuição normal. Enquanto isso, o valor de p para o grupo experimental foi de 0,0099 (isto é, p-value $< \alpha$), revelando que o desempenho acadêmico dos alunos do grupo experimental não segue uma distribuição normal.

Em virtude da distribuição dos dados ser normal em um grupo e anormal em outro, foi utilizado o Teste de Mann-Whitney para testar as hipóteses. Com base no teste, obteve-se o valor de p igual a 0,0900 (isto é, p-value $> \alpha$). Nesse sentido, a hipótese nula não foi rejeita. Assim, pode-se inferir que não existe diferença significativa em termos de desempenho entre estudantes que aprendem com o modelo de *flipped classroom* e com o modelo tradicional. Todos os dados obtidos com o Teste de Mann-Whitney são apresentados na Tabela 9.

Tabela 9 – Teste de Mann-Whitney - Desempenho acadêmico dos alunos

	Grupo controle	**Grupo experimental**
Tamanho da amostra	24	31
Soma dos Postos (Ri)	593,0	947,0
Mediana	5	6,25
U	293,00	
Z (U)	1,3407	
p-valor (unilateral)	0,0900	
p-valor (bilateral)	0,1800	

Fonte: Elaborada pelo autor.

3.2.5 Síntese das Análises

Durante a análise dos dados, uma variável observada foi a motivação. Em conjunto com ela, os quatro itens que constituem a motivação, segundo o modelo ARCS, também foram examinados. Utilizando-se de técnicas de estatística descritiva, os escores de motivação dos alunos do grupo experimental foram superiores aos escores do grupo controle para a categoria de atenção, confiança e satisfação. Mas foram inferiores na categoria de relevância.

Por intermédio dessa análise, em um primeiro momento, acreditou-se que os alunos que aprenderam teste de software com o modelo *flipped classroom* ficaram mais atentos ao adquirir novos conhecimento por meio do modelo. Esses alunos possivelmente tiverem a impressão que se eles se esforçassem teriam sucesso em aprender o conteúdo. E, ao final do processo, naturalmente, ficaram satisfeitos com o modelo pedagógico.

Em contrapartida, os alunos que estudaram com o modelo *flipped classroom* acharam menos relevante esse modelo, em comparação com o tradicional. Acredita-se que esse resultado foi gerado em virtude de os alunos estarem acostumados a estudar com o modelo tradicional, que exige menos participação dos mesmos nas atividades realizadas em sala de aula.

Apesar das análise de medidas de posição nos itens atenção, confiança e satisfação revelarem uma tendência para os escores de motivação geral serem maiores no grupo experimental que no grupo controle, notou-se que as médias ficaram próximas.

Usando *box plots* valores *outliers* foram revelados, mais precisamente nas categorias de atenção (grupo controle), relevância (grupo controle e no grupo experimental) e confiança (grupo controle). Apesar de existirem escores considerados *outliers*, esses dados não foram desconsiderados durante a análise, porque o tamanho da amostra seria minimizado, prejudicando a análise.

Os *box plots* possibilitaram reforçar a ideia que os escores do grupo experimental foram superiores aos do grupo controle, uma vez que essa forma visual de análise revelou as medidas da tendência central. Também foi possível observar a existência de simetria em alguns gráficos, especialmente nos gráficos gerados através dos escores do grupo experimental, indicando que os escores mais frequentes são parecidos com o valor da média e da mediana.

Os *box plots* de motivação, de modo geral, revelaram a existência de *outliers* apenas no grupo controle. Também foi observado que a variância amostral dos escores do grupo experimental foram maiores do que o grupo controle. Assim, foi possível constatar que há uma grande variação entre os escores de motivação de cada estudante que estudou com *flipped classroom*. Esse resultado promoveu uma nova análise, visando entender o grau de motivação dos sujeitos.

A partir disso, foi feita uma categorização da motivação entre os grupos. Com a classificação, foi possível notar que a maioria dos estudantes ficou pouco motivada com as experiências promovidas pelos modelos pedagógicos. Notou-se também que os índices de baixa motivação foram mais frequentes no grupo controle.

Com a finalidade de finalizar a análise a respeito da variável motivação, foram feitos testes de hipótese. Durante esse momento, foram estabelecidas sub-questões de pesquisa para a variável de motivação, que englobaram todas as categorias/itens do modelo ARCS. Por meio da análise, foi possível concluir que os alunos do grupo experimental ficaram mais atentos com o modelo *flipped classroom* em comparação com o tradicional. Apesar dessa descoberta, a motivação, de modo geral, é a mesma entre os grupos. A Tabela 10 sumariza os achados.

Durante a análise descritiva da variável desempenho acadêmico, contatou-se, pre-

Tabela 10 – Resumo das respostas às questões de pesquisa relacionadas a variável motivação

Questão de pesquisa	**Resposta**
QP1 - O modelo *Flipped Classroom* aumenta a motivação do aluno durante aprendizado de teste de software?	Não. O aluno que estuda com *Flipped Classroom* fica tão motivado quanto o aluno que estuda com o modelo tradicional.
QP1.1 O modelo *Flipped Classroom* aumenta a atenção do aluno durante aprendizado de teste de software?	Sim. Os dados indicam que aluno que estuda com *Flipped Classroom* fica mais atento ao estudar teste de software que o aluno que estuda com o modelo tradicional.
QP1.2 O modelo *Flipped Classroom* aumenta a percepção de relevância do aluno durante aprendizado de teste de software?	Não. Não foram encontradas evidências que a percepção sobre a relevância do modelo no aprendizado de teste de software do aluno que estuda com *Flipped Classroom* é maior que a do aluno que estuda com o modelo tradicional.
QP1.3 O modelo *Flipped Classroom* aumenta a percepção de confiança do aluno durante aprendizado de teste de software?	Não. O aluno que aprende com *Flipped Classroom* tem a mesma percepção de que terá sucesso a aprender com esse modelo que o aluno que aprende com o modelo tradicional.
QP1.4 O modelo *Flipped Classroom* aumenta a satisfação do aluno durante aprendizado de teste de software?	Não. Os escores de satisfação indicam que o aluno que aprende com *Flipped Classroom* fica tão satisfeito com o modelo quanto o aluno que aprende com o modelo tradicional.

Fonte: Elaborada pelo autor.

liminarmente, que os alunos do grupo experimental obtiveram pontuações superiores aos alunos do grupo controle. Os *box plots* não revelaram *outliers*, mas indicaram que a medida de tendência central do grupo experimental foi superior a outro grupo. Eles também sugerem que as pontuações dos alunos que estudaram com o modelo tradicional são mais dispersas que as pontuações dos alunos que estudaram com o outro modelo.

Uma análise gráfica foi feita sobre o desempenho dos alunos, por grupo, a partir das intervenções realizadas. Essa análise demonstrou que as intervenções promoveram um acréscimo no desempenho dos alunos. Contudo, no grupo controle alguns alunos obtiveram desempenho muito alto enquanto que outros tiveram o desempenho baixo. Em contraponto, o desempenho dos alunos no grupo experimental não apresentou muita variação.

O teste de hipótese que confrontou o desempenho dos alunos de estudaram o conteúdo de teste de software com o desempenho dos alunos que estudaram esse mesmo conteúdo com o modelo tradicional revelou que não há diferença entre esses desempenhos.

Nesse sentido, presume-se que o desempenho dos alunos que estudam com *flipped classroom* é equivalente ao desempenho dos estudantes que estudam com o modelo tradicional.

Em relação ao desempenho acadêmico, resultados similares foram obtidos por Bergmann e Sams (2012), em que os autores mencionam que os alunos de química do 8º ano do ensino fundamental não obtiveram índices maiores de desempenho que os alunos que haviam estudante com o modelo tradicional. Esses autores acreditam que de modo geral, o modelo *flipped classroom* tende a deixar os alunos mais engajados, mas não promove um aumento no desempenho dos alunos, se comparado ao modelo tradicional.

3.2.6 Ameaças à Validade

Todo estudo experimental está sujeito a conter ameaças as quais podem invalidar o trabalho conduzido. Conscientes dessa informação, durante o planejamento foram estabelecidas estratégias para mitigar a presença de ameaças que comprometem a pesquisa. Assim, esta seção busca apresentar as ameaças à validade desse estudo. Antes de apresentá-las, é importante destacar que elas foram categorizadas conforme recomendado por Cook e Campbell (1979), classificando-as em validade de conclusão, validade interna, validade de constructo e, por último, validade externa.

Validade de conclusão: Essa categoria está relacionada a obtenção de conclusões corretas sobre o fenômeno analisado, a partir dos resultados. Esses resultados podem ser influenciados por testes estatísticos selecionado e utilizado na análise dos resultados. No que se refere à estratégia utilizada para mitigar essa ameaça, os testes estatísticos foram pré-definidos seguindo as sugestões da literatura especializada.

Para a definição, primeiramente foi analisado a normalidade da distribuição dos dados e o tipo de amostra (no contexto desse estudo não foram utilizadas amostras pareadas), na sequência o teste estatístico foi definido. Adicionalmente, para o nível de significância (alfa), foi estabelecido um valor amplamente aceito pela literatura acadêmica da área.

Outro item que se refere à validade de conclusão é a confiabilidade das medidas. De acordo com Wohlin *et al.* (2012), formulários mal projetados e/ou questões mal formuladas (escritas) podem prejudicar a validade da pesquisa. Para minimizar os riscos à validade neste aspecto os formulários utilizados foram analisados previamente pelo pesquisador, por dois estudantes de Pós-Graduação (uma mestranda em Computação Aplicada a Educação e uma doutoranda em engenharia de software, com pesquisa na área de teste de software) e dois especialistas da área.

Por último, o ambiente de operação e irrelevâncias aleatórias nesse ambiente (*e.g.*, barulho produzido pelos alunos, interrupções geradas por terceiros) podem ameaçar

a validade do experimento. Para mitigar esse tipo de ameaça, os alunos foram instruídos para não fazer barulho no momento em que a professora estava solucionando dúvidas, explicando o conteúdo (para alunos do grupo controle) e deveriam evitar conversas quando eles estivessem respondendo algum tipo de formulário.

Também foi adotada uma estratégia para começar as atividades em sala de aula somente após a entrada de todos os alunos, visando evitar quaisquer tipos de tumulto. Por fim, as salas de aulas utilizadas no experimento não ficam próximas a corredores com fluxo contínuo de alunos, evitando assim o alvoroço provocado por terceiros.

Validade interna: A relação causa-efeito é uma questão chave em um estudo experimental. Desse modo, a validade interna é uma categoria que busca retratar se as mudanças observadas podem ser atribuídas ao tratamento (causa) e não a outros fatores.

No contexto desse experimento, alguns fatores podem oferecer algumas ameaças ao trabalho, são elas: número de materiais disponibilizados aos alunos do grupo experimental, qualidade desses materiais e a aula ministrada pela professora aos alunos do grupo controle. Com o propósito de mitigar essas ameaças, algumas atitudes foram tomadas.

Em relação à quantidade de materiais didáticos, inicialmente, eles foram escolhidos em conjunto com dois especialistas. Eles foram selecionados de modo que o aluno pudesse escolher qual o formato de material de sua preferência. Assim, apesar de existir diversos materiais, os alunos foram avisados que eles poderiam selecionar o material didático de sua preferência.

A qualidade dos materiais didáticos, por sua vez também foi ponderada. Apesar de não ter sido aplicado um questionário previamente com alunos para eles indicarem a qualidade do material utilizado, antes da disponibilização o material foi selecionado pelo pesquisador em conjunto com os especialistas.

Em relação à aula ministrada aos alunos do grupo controle, para mitigá-la, uma professora com mais de 15 anos de experiência em teste de software, que já ministrou mais de 5 vezes o conteúdo abordado foi quem ministrou a aula para os alunos desse grupo. Essa professora utilizou um conjunto de slides que também havia sido disponibilizado aos alunos do grupo experimental.

Um outro fator que está estritamente relacionado com a quantidade de materiais didáticos é o cansaço do participante. Em razão do aluno estudar com todos os materiais disponibilizados, um cansaço pode ter sido provocado. Como solução para não ocasionar cansaço nos estudantes, eles foram avisados que não deveriam utilizar

todos os materiais, selecionar apenas um ou dois formatos, de acordo com suas preferências.

Outra ameaça à validade é a mortalidade, em que os sujeitos participantes da pesquisa desistem durante a condução do experimento. Para mitigar essa ameaça, no início do experimento os alunos foram avisados que poderiam escolher participar das iniciativas, as quais seriam consideradas optativas na disciplina e não seriam acrescentadas às avaliações.

Validade de constructo: A categoria de validade de constructo está estritamente relacionada ao planejamento (*design*) do experimento. Para mensurar o desempenho e a motivação, foi definido apenas um instrumento para cada variável. No caso do desempenho, um teste com questões múltipla-escolha foi adotado no estudo. Já em relação a motivação, o IMMS com escalas Likert foi o único formulário utilizado.

Em relação ao teste com questões múltipla escolha, acredita-se que o ideal seria promover diferentes atividades para mensurar o desempenho acadêmico dos alunos. No entanto, para não prejudicar o andamento da disciplina de engenharia de software foi definido que seria a estratégia mais ágil para se conseguir mensurar o desempenho dos alunos. Além disso, dentre as avaliações existentes questionários múltipla escolha são opções adotadas por muitas professores em instituições de ensino.

Em relação ao instrumento para mensurar a motivação, foi utilizado o IMMS, questionário amplamente utilizado na literatura. De acordo com Santana, Araújo e Bittencourt (2018), apesar dos constructos do IMSS estarem mais associadas à materiais de aprendizagem, eles oferecem uma base para avaliação de iniciativas que envolvem instrução. Além disso, muitos outros instrumentos psicométricos que analisam a motivação são mais direcionados a formatos específicos, como ambientes de aprendizagem.

Outro aspecto que pode ameaçar à validade do estudo refere-se a memorização das questões do instrumento que foi destinado a mensuração do desempenho acadêmico. O instrumento foi aplicado antes dos estudantes participarem dos estudos com os diferentes modelos pedagógicos e, após a iniciativa. Para mitigar a ameaça, a ordem de apresentação das questões do teste e de suas respectivas alternativas foram alteradas.

Validade externa: A categoria de validade externa está associada às dificuldades de generalizar os resultados que foram obtidos para o contexto real. No contexto desse experimento, essa ameaça foi mitigada a partir da seleção de um contexto acadêmico. Além disso, os participantes da pesquisa eram alunos da disciplina de engenharia de software, que de acordo com Paschoal e Souza (2018b) é a disciplina em que a maioria dos professores de Computação ensina teste de software.

Considerando o objetivo do trabalho, o conteúdo de teste de software foi utilizado, mais especificamente a técnica de teste funcional. Acredita-se que apesar de um ambiente real ter sido utilizado, não é possível generalizar os resultados para outras técnicas e critérios de teste de software. Pode ser que os resultados encontrados sejam diferentes ao ser considerado alguma outra técnica e/ou critério.

3.2.7 Apresentação e Empacotamento

Após a condução do estudo experimental, de acordo com Wohlin *et al.* (2012) é importante disponibilizar os resultados obtidos. Essa disponibilização pode ser feita por meio de artigos, relatórios técnicos ou materiais didáticos.Alguns resultados deste experimento foram publicadas em Paschoal, Silva e Souza (2017).

Visando contribuir com o desenvolvimento de novas pesquisas na área, os formulários, materiais utilizados na instrução dos participantes e os dados coletados foram disponibilizados na forma de pacote de laboratório de acesso público. O pacote de laboratório para este experimento pode ser obtido no seguinte endereço: <http://bit.ly/2FSw5qj>.

A disponibilização do material é importante porque pode facilitar replicações do estudo e comparações dos modelos avaliados nesse estudo com outros modelos propostos pela literatura. Além disso, o material pode ser utilizado para auxiliar no ensino de teste de software.

3.3 Segundo Experimento

O primeiro experimento indicou resultados semelhantes para o modelo *flipped classroom* e o tradicional. Nesse experimento apesar do desempenho do aluno em atividades ter sido avaliado, não foi mensurado o quanto o aluno aprendeu a partir do modelo. Conforme Kong (2014), é importante reconhecer a eficácia do modelo no contexto do ganho de conhecimento do aluno.

No estudo de He *et al.* (2016), é descrito que a avaliação da eficácia do modelo deve ponderar diferentes variáveis, não apenas o desempenho e/ou motivação. Nesse sentido, o autor retoma o propósito do modelo *flipped classroom*. Em aplicações de *flipped classroom* busca-se aproveitar o tempo de sala de aula para uma explicação mais profunda, prática e uso produtivo do conhecimento, uma vez que no modelo tradicional as preleções feitas pelo professor consomem quase todo o tempo de aula. Nessa perspectiva, no modelo tradicional pouco tempo resta para os estudantes trabalharem com atividades práticas de fixação em sala de aula.

A efetiva implementação do modelo *flipped classroom* exige que os estudantes se preparem para as aulas no momento “antes das aula”, o que pode ocasionar um aumento

na carga de trabalho do aluno. Gannod, Burge e Helmick (2008) apresentam alguns argumentos sobre suas crenças de que os alunos que estudam com *flipped classroom* precisam dedicar mais tempo para se preparar para as aulas. Com base nisso, emerge-se a necessidade de analisar a carga de trabalho gerada pelo modelo, especialmente no conteúdo de teste de software. He *et al.* (2016) reforçam descrevendo que os pesquisadores de *flipped classroom* devem medir o esforço de estudo sempre que possível, pois se há um esforço demasiado por meio desse modelo, ele pode se tornar menos atrativo do ponto de vista pedagógico.

No estudo experimental apresentado na seção anterior, a relevância acerca dos modelos pedagógicos foi analisada. Durante a análise da categoria de relevância do IMMS, a estatística descritiva dos dados revelou que os escores do grupo experimental nessa categoria são menores que os escores do grupo controle. Assim, presumiu-se que o grupo de alunos que estudou com o modelo *flipped classroom* não achou esse modo relevante para o aprendizado de teste de funcional. Com base nisso, torna-se necessário averiguar as percepções dos alunos sobre o modelo, isto é, verificar se eles realmente preferem ou não o modelo *flipped classroom*.

A partir desse contexto, esta seção tem a finalidade de apresentar um estudo experimental conduzido com o objetivo de investigar o impacto de *flipped classroom* sobre a aquisição e retenção de conhecimento, o impacto do modelo no tempo de estudo (instrução) e nas percepções dos alunos.

3.3.1 Definição

A primeira atividade do experimento foi a definição do escopo do estudo. Assim, as próximas subseções irão apresentar o estabelecimento do objetivo, o objeto que está sob investigação, o foco qualitativo que é adotado no experimento, a perspectiva e o contexto em que este experimento foi realizado.

Objetivo

O objetivo é investigar o modelo *flipped classroom* (tratamento) e seus benefícios para o aprendizado de teste de software. O experimento busca entender os efeitos do uso do modelo no ensino de teste funcional, dando atenção a aprendizagem imediata, a aprendizagem efetiva a partir de atividades prática e a retenção de conhecimento, o tempo consumido pelo aluno durante a instrução e as percepções do aluno que é exposto ao modelo *flipped classroom*. Essa investigação é necessária porque os efeitos do modelo não foram explorados nos aspectos mencionados e, conforme a literatura, merecem ser considerados.

Seguindo o modelo apresentado no estudo de Basili e Rombach (1988), definiu-se

o objetivo do estudo: **analisar** o modelo *flipped classroom*, **com o propósito** de verificar seu impacto como modelo de ensino, **com respeito ao** aprendizado, retenção de conhecimento, tempo de estudo e nas percepções do aluno, **do ponto de vista** do pesquisador, **no contexto de** estudantes de Engenheira de Software, estudando o conteúdo de teste de software.

Objeto

O objeto que está sob investigação é o modelo *flipped classroom*. Esse modelo é aplicado no contexto de ensino de teste de software.

Foco Qualitativo

O foco qualitativo do estudo é mensurar o impacto do modelo na aprendizagem do aluno (aprendizagem imediata, aprendizagem efetiva e conhecimento retido) e na carga de trabalho do aluno (tempo gasto durante a instrução). Tem-se também como foco analisar as percepções do aluno sobre o modelo, após ele ter contato com *flipped classroom*.

Perspectiva

A perspectiva desta pesquisa é feita sob o ponto de vista acadêmico, de um aluno de pós-graduação e da professora orientadora. Assim, o experimento foi planejado e executado em um ambiente acadêmico.

Contexto

Este experimento foi conduzido no primeiro semestre de 2018, com alunos do 7º semestre do curso de Engenharia de Computação, oferecido em parceira estabelecida entre o Instituto de Ciências Matemática e de Computação e a Escola de Engenheira de São Carlos, nas dependências da Universidade de São Paulo. Os alunos participantes da pesquisa estavam matriculados na disciplina de engenharia de software[3]. Na ocasião, os alunos dessa disciplina haviam começado a estudar o conteúdo de teste de software, mas não haviam sido introduzidos aos critérios e as técnicas de teste. Assim, o conteúdo abordado no experimento foi a técnica de teste funcional e o critério particionamento em classes de equivalência.

3.3.2 Planejamento

Seleção do Contexto

Este experimento provê um processo *offline* em que os participantes são entrevistados em alguns momentos específicos durante o experimento. Os participantes, por sua

[3] Mais informações sobre a disciplina estão disponíveis em: < http://bit.ly/2MpPCze >

vez, são alunos de graduação, com primeiro contato com a área de teste de software. Em relação à dimensão de realidade, o experimento pode ser considerado um estudo real, que utiliza conteúdo, atividades, e o seu impacto é mensurado por exercícios que são comumente utilizados em disciplinas de Engenharia e/ou teste de software. Outro aspecto que contribui para a caracterização como real é que os alunos participantes estudaram esse conteúdo em uma disciplina legítima, oferecida pelos institutos da Universidade de São Paulo. Isso é importante para esclarecer a generalidade, uma vez que o ambiente é real, os dados podem ser generalizados para situações similares, em que o mesmo conteúdo é abordado.

Definição das Hipóteses

Em virtude desse experimento ser um desdobramento do primeiro, ele também busca contribuir com a construção de evidências para responder a primeira questão de pesquisa abordada nesta dissertação, a saber: "É viável ensinar conteúdos de teste de software com o modelo pedagógico *flipped classroom*?". Nessa perspectiva, este experimento investiga algumas questões específicas. Para cada uma delas, hipóteses de pesquisa foram pré-definidas:

Primeira Questão de Pesquisa O aprendizado de teste de software por meio do modelo *flipped classroom* é maior do que o aprendizado por meio do modelo de ensino tradicional?

- **Hipótese Nula:** Não existe diferença entre o aprendizado por *flipped classroom* e o aprendizado por meio do modelo tradicional.

  H_0: Aprendizado $_{\text{Flipped Classroom}}$ = Aprendizado $_{\text{Tradicional}}$
- **Hipótese Alternativa:** Existe diferença entre o aprendizado por *flipped classroom* e o aprendizado por meio do modelo tradicional.

  H_1: Aprendizado $_{\text{Flipped Classroom}}$ $\neq$Aprendizado $_{\text{Tradicional}}$

Segunda Questão de Pesquisa Os alunos de teste de software que estudam com *flipped classroom* passam mais tempo estudando do que os alunos que estudam com o modelo tradicional?

- **Hipótese Nula:** Não existe diferença entre o tempo gasto pelo aluno que estuda com *flipped classroom* e o tempo gasto pelo aluno que estuda por meio do modelo tradicional.

  H_0: Carga de Trabalho $_{\text{Flipped Classroom}}$ = Carga de Trabalho $_{\text{Tradicional}}$
- **Hipótese Alternativa:** Existe diferença entre o tempo gasto pelo aluno que estuda com *flipped classroom* e o tempo gasto pelo aluno que estuda por meio do modelo tradicional.

H_1: Carga de Trabalho $_{\text{Flipped Classroom}}$ $\neq$Carga de Trabalho $_{\text{Tradicional}}$

Terceira Questão de Pesquisa Os alunos de teste de software que estudam com *flipped classroom* preferem esse modelo pedagógico em detrimento ao modelo tradicional?

- **Hipótese Nula:** A preferência pelo modelo *flipped classroom* é equivalente a preferência pelo modelo tradicional.

  H_0: Preferência $_{\text{Flipped Classroom}}$ = Preferência $_{\text{Tradicional}}$

- **Hipótese Alternativa:** A preferência pelo modelo *flipped classroom* é diferente a preferência pelo modelo tradicional.

  H_1: Preferência $_{\text{Flipped Classroom}}$ $\neq$Preferência $_{\text{Tradicional}}$

Seleção dos Sujeitos

A seleção de estudantes para participar deste experimento foi baseada em conveniência. Nesse sentido, alunos do 7º semestre do curso de Engenharia de Computação, matriculados na disciplina de engenharia de software foram convidados a partir do experimento. No primeiro semestre de 2018, a disciplina contou com a participação de um total de 41 alunos. Esses alunos foram divididos em dois grupos, de maneira randômica. 21 alunos foram designados ao grupo controle (aluno que estudaram teste de software com o modelo tradicional de ensino) e 20 ao grupo experimental (aluno que estudaram teste de software com o modelo *flipped classroom*). Esses alunos não possuíam experiência anterior com a área de teste de software.

Seleção das Variáveis

As variáveis independentes consideradas neste experimento são as mesmas descritas na Seção 3.2.2. As variáveis dependentes mudam, em razão dos outros aspectos que são avaliados e mensurados no decorrer deste experimento. Diante disso, as variáveis dependentes serão apresentadas na próxima seção.

Variáveis Dependentes

Neste experimento, as seguintes variáveis dependentes (variáveis de saída) são observadas:

- **Aprendizado:** Essa variável refere-se ao aprendizado adquirido pelo estudante a partir das intervenções proporcionadas pelas especificidades de cada um dos modelos. Para mensurar o aprendizado, foram considerados algumas argumentações descritas no estudo de Lyra *et al.* (2016), em que o aprendizado é observado em diferentes estágios, durante o ensino, são eles: aprendizado imediato (representado por $\Delta 1$),

aprendizado a partir de atividades ativas (representado por $\Delta 2$) e conhecimento retido (representado por $\Delta 3$).

$\Delta 1$ refere-se a quanto um estudante aprende imediatamente após a instrução com o modelo pedagógico.

$\Delta 2$ refere-se a quanto um estudante melhorou o seu conhecimento após a realização de atividades ativas.

$\Delta 3$ refere-se ao conhecimento retido pelo estudante ao final do experimento.

- **Tempo de instrução:** Essa variável refere-se ao tempo em minutos que cada aluno "gasta" para estudar o conteúdo, considerando apenas o período que ele utiliza para se instruir. Essa especificidade é importante porque estudos discutem que o modelo promove uma carga de trabalho maior, induzido pelas atividades obrigatórias no momento "antes da aula".

- **Preferência pelo modelo pedagógico:** Essa variável refere-se a quanto um aluno acostumado com o modelo tradicional de ensino gosta do modelo pedagógico *flipped classroom* ao ter contato com o modelo pela primeira vez.

Descrição da Instrumentação

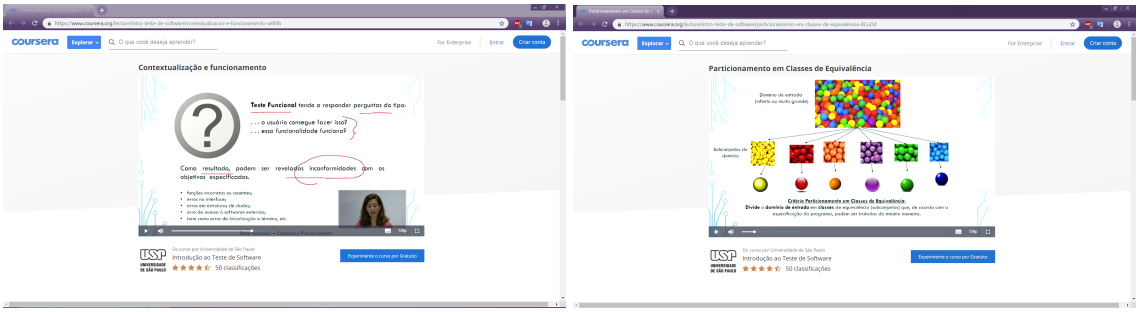
Esta seção tem o propósito de apresentar os instrumentos preparados e utilizados no estudo experimental. Alguns materiais foram utilizados para instruir os estudantes, outros para coletar os dados usados na análise deste trabalho. Além desses recursos, a versão institucional do ambiente virtual de aprendizagem Moodle foi empregada, visando disponibilizar os materiais de estudo aos alunos do grupo experimental.

Diferentemente do primeiro experimento em que foram disponibilizados materiais didáticos instrucionais em diferentes tipos de formatos (*e.g.* vídeo, slides, artigo), neste experimento foram selecionados apenas vídeos. Ao total foram disponibilizados duas vídeo aulas, uma sobre uma visão geral de teste funcional e a outra sobre o critério particionamento em classes de equivalência. Esses vídeos foram selecionados de cursos MOOC sobre teste de software e junto totalizavam 15 minutos e 49 segundos.

Durante a seleção dos vídeos, 4 pesquisadores fizeram a revisão dos mesmos, dois professores com mais de 15 anos de experiência na área de teste de software, um doutorando na área de engenharia de software e o aluno de mestrado em teste de software. A Figura 14 apresenta uma captura de tela dos vídeos selecionados. Salienta-se que para a aula tradicional, o conjunto de slide utilizado no primeiro experimento foi adotado.

Com o propósito de mensurar o aprendizado, foram elaborados três testes (pré-teste, teste intermediário e pós-teste). Cada um deles é constituído por um conjunto de questões múltipla escolha. O pré-teste é um questionário composto por dez questões que

Figura 14 – Vídeos selecionados para a instrução invertida



Fonte: Elaborada pelo autor.

avalia o conhecimento prévio dos alunos. Para cada questão, os alunos devem selecionar apenas uma opção de resposta dentre as cinco que são disponíveis. Esse teste será utilizado como valor inicial para mensurar o quanto um aluno ganhou ou reteve de conhecimento no decorrer do experimento.

O teste intermediário é questionário de conhecimento que possui variações das dez questões do pré-teste. Esse instrumento é aplicado após o aluno ser instruído sobre o conteúdo de teste, por meio dos modelos pedagógicos. Nesse sentido, ele busca identificar o conhecimento adquirido a partir da instrução. Por fim, o pós-teste também contém uma variação das dez questões e é aplicado após os alunos realizarem atividades práticas sobre o conteúdo e solucionarem as dúvidas que surgiram durante a realização da atividade. O pós-teste é utilizado para mensurar o conhecimento retido após toda a iniciativa. Salienta-se que as questões foram retiradas de teste de certificação da área de teste de software e adaptadas pelos pesquisadores deste estudo.

O tempo gasto pelo aluno durante a instrução não é trivial de ser mensurado. No estudo de He *et al.* (2016), os autores solicitam aos seus estudantes (alunos de Química e Biologia) que anotem o tempo que cada um utilizou no momento de estudo e, no início da aula, aplicou um questionário visando coletar essas informações. Com base nisso, apesar de não se ter uma manipulação direta, têm-se um subsídio para analisar o tempo consumido pelo aluno do modelo *flipped classroom*. No contexto desse experimento, essa estratégia também foi adotada. Assim, um questionário foi formulado visando coletar informações relacionadas ao tempo que cada aluno levou para se instruir/preparar para a aula. Uma descrição completa de como é feita a mensuração do tempo de instrução será descrita na próxima seção.

Com o intuito de verificar se os alunos que tiveram uma experiência de aprender teste de software com o modelo *flipped classroom* preferem esse modelo em detrimento ao tradicional, bem como ter uma visão geral sobre as percepções desses alunos, foi utilizado um questionário específico. O questionário em questão foi proposto e avaliado por He *et al.* (2016) e possui questões sobre o conteúdo, sobre o material didático, instrução fora

da sala de aula e instrução em sala de aula. Esse instrumento foi traduzido do idioma inglês para o português, uma vez que todo o material usado no experimento foi preparado nesse idioma, e avaliado por quatro pesquisadores (os dois doutores, o doutorando e o mestrando).

Por fim, o último instrumento desenvolvido foi o termo de consentimento. O propósito é deixar o aluno ciente sobre os seus deveres e direitos. Esse instrumento e todos os demais descritos ao longo desta seção estão disponíveis no pacote de laboratório, discutido no decorrer da Seção 3.3.7.

3.3.3 Operação

O estágio de operação envolve a preparação dos participantes para execução do experimento, organização do material usado no experimento, o planejamento e a implementação dos modelos pedagógicos na disciplina de engenharia de software. Também consiste na execução do experimento, em conformidade com o planejamento previamente estabelecido.

Preparação

Após a definição do experimento e respectivo planejamento, a professora responsável por ministrar a disciplina de engenharia de software no semestre 2018/1 foi contatada. Nesse momento, a definição e o planejamento do experimento foram apresentados. A professora aceitou que a condução do mesmo ocorresse no contexto de sua disciplina. Nesse momento, foram discutidos o número de aulas necessárias para a realização do experimento, os ajustes necessários no cronograma da disciplina para que o experimento não atrapalhasse o andamento da mesma. Ademais, foi estabelecida uma estratégia para não prejudicar o aprendizado dos alunos na disciplina.

A tática estimulada para não prejudicar o desenvolvimento da disciplina, nem o aprendizado dos alunos, foi estabelecida de modo que todos os estudantes da disciplina participassem das atividades. Para tanto, os alunos deveriam ser notificados sobre o experimento e aqueles que não desejassem disponibilizar os dados para análise deveriam realizar as atividades previstas, mas elas não seriam consideradas na análise. Assim, o experimento não foi considerado uma atividade avaliativa e mesmo que os alunos optassem por não disponibilizar os seus dados, eles continuariam aprendendo o conteúdo, porém não participariam dos testes de conhecimento.

A professora responsável pela disciplina e um doutorando que realizou estágio docente na disciplina auxiliaram os pesquisadores na definição e revisão dos materiais didáticos e nas atividades que foram selecionadas para o experimento. Esses profissionais, especialistas na área de engenharia de software também contribuíram com a definição e

revisão das questões que compõe os testes de conhecimento (pré-teste, teste intermediário e pós-teste).

Os testes de conhecimento foram desenvolvidos de forma cuidadosa, constituídos por questões retiradas de testes de certificação internacional, buscando avaliar de forma efetiva o conhecimento dos alunos no momento em que eles realizam os testes. Durante a construção desses testes, foi dada atenção à prática do critério particionamento em classes de equivalência. Alguns exemplos de questões adaptadas do exame estão disponíveis na Tabela 11. Conforme é possível observar as questões foram modificadas, mas a estrutura delas continua a mesma. Em alguns casos, as respostas não são as mesmas.

Durante a preparação da operação, a versão institucional do ambiente Moodle foi configurada para apresentar os materiais didáticos disponíveis para o estudo. Essa configuração foi realizada por meio das ferramentas de criação de grupos do Moodle em conjunto com a opção de ocultar materiais aos participantes que não estão em um determinado grupo.

O questionário desenvolvido no estudo de He *et al.* (2016), que busca coletar informações sobre as percepções dos alunos, foi traduzido e adaptado ao contexto da temática trabalho. Assim questões como "estou muito interessado no conteúdo e na área deste curso" foram adaptadas para "estou muito interessado no conteúdo de teste funcional". A escala e estrutura gramatical e, consequentemente, o sentido da pergunta permanecem.

A divisão dos estudantes em dois grupos provocou o surgimento da necessidade de duas salas de aula. Uma delas reservada a instrução dos alunos do grupo experimental com o modelo *flipped classroom* e a outra destinada aos alunos do grupo controle, que teriam que estudar o conteúdo de teste com o modelo tradicional de ensino.

Por fim, o último estágio para a preparação da operação consistiu em definir como ocorreria a aula tradicional. Nesse sentido, foi definido que a professora responsável por ministrar a disciplina realizaria preleções sobre o conteúdo, utilizando os slides, de uma maneira tradicional. Ela ministraria o conteúdo planejado em um período de tempo pré-determinado, solucionaria as dúvidas dos alunos e disponibilizaria uma atividade. Após a aula, a professora disponibilizaria o conjunto de slides aos alunos do grupo controle.

Em relação à aula com *flipped classroom*, ficou definido que os alunos deveriam acessar o ambiente Moodle, assistir aos vídeos, anotar o tempo gasto para o estudo e as dúvidas que ficaram ao estudar o conteúdo. Em sala de aula, esses alunos teriam um momento para solucionar as dúvidas, realizar uma atividade em duplas com seus pares e com a supervisão do professor da disciplina.

Tabela 11 – Exemplo de questões que constituem os testes de conhecimento

	Descrição
Questão original	Given the following specification, which of the following values for age are in the SAME equivalence partition? If you are less than 18, you are too young to be insured. Between 18 and 30 inclusive, you will receive a 20% discount. Anyone over 30 is not eligible for a discount. (A) 17, 18, 19 (B) 29, 30, 31 (C) 18, 29, 30 (D) 17, 29, 31
Pré-teste	Considerando a seguinte especificação: Se você tem menos de 18 anos, é jovem demais para ser segurado. Entre 18 e 30 anos, você receberá um desconto de 20%. Qualquer pessoa com mais de 30 anos não é elegível para um desconto. Quais dos seguintes valores para idade estão na mesma partição de equivalência? (A) 17, 18, 19 (B) 29, 30, 31 (C) 18, 29, 30 (D) 17, 29, 31 (E) Nenhuma das alternativas anteriores
Teste intermediário	Considerando a seguinte especificação: Se você tem menos de 18 anos, é jovem demais para ser segurado. Se você tem mais de 60 anos, você receberá um desconto de 50%. Qualquer pessoa com menos de 60 anos e mais de 17 anos não é elegível para um desconto. Quais dos seguintes valores para idade estão na mesma partição de equivalência? (A) 17,18,19 (B) 18,19,20 (C) 18,60,17 (D) 59,60,61 (E) Nenhuma das alternativas anteriores
Pós-teste	Considerando a seguinte especificação: Se você tem menos de 18 anos, é jovem demais para ser segurado. Se você tem mais de 60 anos, você receberá um desconto de 50%. Qualquer pessoa com menos de 60 anos e mais de 17 anos não é elegível para um desconto. Quais dos seguintes valores para idade estão na mesma partição de equivalência? (A) 17, 18, 19 (B) 18, 19, 20 (C) 18, 60, 17 (D) 59, 60, 61 (E) Nenhuma das alternativas anteriores

Fonte: Elaborada pelo autor.

Execução

Este experimento foi executado no decorrer de duas semanas, em quatro encontros da disciplina. O primeiro, o segundo e o quarto encontro ocorreram em uma quarta-feira e o terceiro encontro em uma sexta-feira (entre o primeiro e o segundo momento, um

feriado ocasionou o cancelamento da aula na sexta-feira, fazendo com que o segundo encontro do experimento ocorresse uma semana após o primeiro). Neste experimento foram utilizadas 4 encontros, porque foram propostas atividades de fixação de conteúdo no decorrer do experimento. Esta seção visa apresentar as atividades do experimento realizadas no decorrer dessas aulas. A descrição será feita conforme os cinco estágios que constituem o experimento.

Estágio 1: No primeiro encontro a professora da disciplina informou aos alunos que utilizaria um modelo pedagógico diferente do tradicional no ensino de teste funcional. Ela explicou aos alunos que o modelo em questão se trata do *flipped classroom* e detalhou o modelo. Ela falou que dividiu a turma de maneira randômica em dois grupos. Eles foram notificados que todos realizariam as mesmas atividades, mas que elas não seriam avaliativas. Foi informado que os alunos responderiam formulários durante os quatro encontros, mas aqueles que não quisessem disponibilizar os seus dados, estariam livres de entregá-los. Todavia, todos teriam que participar, pois a disciplina não teria outro momento para abordar o conteúdo de outra forma. Uma explicação sobre as atividades foi entregue em formato de termo de consentimento e os alunos deveriam preencher.

Após o preenchimento do termo de consentimento, a professora da disciplina explicou que antes da atividade ela gostaria de saber o conhecimento deles sobre o conteúdo que seria trabalhado durante o experimento. Para tanto, ela entregou os testes de conhecimento (pré-teste) e pediu aos alunos que respondessem de maneira individual e sem consulta a fontes de busca (*e.g.*, Internet, Livros).

Na sequência, os alunos do grupo controle foram liberados da aula. Isso foi necessário porque os alunos do grupo experimental precisariam ser notificados sobre os materiais disponíveis no Moodle e que eles deveriam estudar o conteúdo por meio dos vídeos, anotar o tempo que eles levaram para estudar e fazer anotações das dúvidas que foram surgindo no decorrer do experimento. Esse grupo de alunos também foi informado que não poderiam expor aos seus colegas que eles teriam que estudar para a próxima aula, ficando avisados que eles não poderiam também enviar os materiais ao outro grupo.

Estágio 2: Esse estágio compreende as atividades realizadas pelos alunos que estudaram com o modelo *flipped classroom*. Os alunos "antes da aula" tiveram que acessar os vídeos e assistir as preleções que haviam sido selecionadas e disponibilizas anteriormente. No vídeo há uma explicação teórico-prática e exemplos de aplicação do critério particionamento em classes de equivalência. Após assistir os vídeos, "durante a aula" do segundo encontro, os alunos desse grupo tiveram 15 minutos para solucionar as dúvidas geradas no momento de estudo ("antes da aula"). Na ocasião,

houve uma discussão sobre a interpretação dos alunos sobre os vídeos e uma breve revisão. Após a resolução de dúvidas os alunos tiveram que responder o formulário sobre suas percepções sobre o modelo estudado e indicar o tempo que levaram para estudar o conteúdo "antes da aula".

Depois dos alunos responderem o formulário de percepção, um teste de conhecimento (teste intermediário) foi aplicado. Foi explicado que o propósito era reconhecer o quanto os alunos conseguiram adquirir de conhecimento sobre o conteúdo por meio da intervenção invertida e resolução de problemas/dúvidas. Os alunos não poderiam fazer consultas a fontes de informação/conhecimento e deveriam responder o formulário individualmente.

Estágio 3: Ao mesmo tempo que o segundo estágio ocorria em uma sala de aula, em outra os estudantes selecionados para aprender teste de software com o modelo tradicional assistiam a uma aula tradicional. A aula teve duração aproximada de um período (50 minutos), em que 30 minutos foram destinados exclusivamente para o ensino da técnica e o critério abordado no estudo e resolução de dúvidas, 20 minutos para os estudantes fazer o teste de conhecimento (teste intermediário).

Estágio 4: Após uma aula sobre o conteúdo feita pelo grupo controle e a resolução de dúvidas do grupo experimental, os estudantes tiveram a oportunidade de realizar uma atividade em dupla. Alunos do grupo experimental só poderiam faz duplas com alunos desse mesmo grupo. Do mesmo modo, alunos do grupo controle também deveriam fazer duplas com alunos do mesmo grupo. Os alunos realizaram as atividades e foram solucionando as dúvidas na medida que elas iam surgindo.

Salienta-se que apesar da prática em sala de aula ser menos executada no modelo tradicional que em modelos direcionados a aprendizagem ativa, foi necessário oportunizar a atividade em sala de aula para que os alunos solucionassem sem trapaça, pedir para os veteranos realizar, ou qualquer atividade similar. A proposta dessa atividade foi oportunizar aos alunos, independente do grupo, um momento para reforçar o conhecimento adquirido, realizando uma atividade prática.

Estágio 5: O quinto estágio consiste nas atividades realizadas no quarto encontro. Nesse momento, os alunos do grupo controle e experimental se encontraram em uma mesma sala de aula. Na ocasião, eles foram convidados a responder mais um teste de conhecimento (pós-teste). De maneira semelhante aos testes anteriores, eles não poderiam realizar qualquer tipo de consulta e deveriam fazê-lo individualmente. Esse teste foi aplicado com o propósito de averiguar o quanto os alunos conseguiram aprender a partir das iniciativas realizadas, em cada um dos modelos.

Após a execução do experimento, os dados coletados foram tabulados e analisados. A próxima seção apresenta as análises realizadas.

3.3.4 Análise dos Resultados

Esta seção apresenta os resultados obtidos a partir da condução do experimento. Para tanto, inicialmente é descrita uma caracterização dos sujeitos que participaram do experimento. Na sequência, visando entender se o aprendizado de teste de software por intermédio do estudo com *flipped classroom* é maior que o aprendizado tradicional, é apresentada uma análise considerando os acertos dos estudantes nos testes de conhecimento e o cálculo de $\Delta 1$, $\Delta 2$ e $\Delta 3$.

Após a apresentação da análise do aprendizado, é feita uma descrição sobre a carga de trabalho promovida no aluno que estuda com o modelo *flipped classroom*. Os resultados são comparados com o tempo que o aluno do modelo tradicional leva durante a instrução ao estudar o conteúdo. Por último, é apresentada uma análise sobre as percepções dos alunos que tiveram contato com o modelo *flipped classroom*.

Em virtude do número limitado de participantes no experimento, não foi possível utilizar testes de inferência estatística. Nesse sentido, os dados foram analisados com técnicas de estatística descritiva (medidas de posição e metidas de dispersão). Assim, diferentemente do primeiro experimento que possuía seções específicas para cada tipo de análise (*i.e.*, estatística descritiva e testes de hipótese), no segundo experimento cada variável observada possui uma única seção.

Caracterização dos Participantes

No início do experimento, todos os estudantes da disciplina foram convidados a participar do estudo, disponibilizando dados no decorrer do experimento através de testes de conhecimento e formulários. Apesar de todos os alunos participarem da iniciativa proposta na disciplina, alguns estudantes optaram por não disponibilizar os seus dados. Isso promoveu uma redução no número de dados amostrais para a análise. Além disso, alguns estudantes não participaram de todas as iniciativas promovidas no decorrer do experimento. Por consequência disso, formulários não foram estregues, alguns alunos não realizaram testes de conhecimento, comprometendo a amostragem.

Foi feita uma revisão em todos os testes, formulário, no registro de participação dos alunos nas atividades do experimento e, na sequência, os dados de alunos que estavam incompletos (*e.g.*, aluno não entregou o teste intermediário, aluno esqueceu de responder alguma questão no formulário do *feedback*, dentre outras) foram descartados. Assim, para a análise restaram apenas dados amostrais de 27 estudantes. Desse total, 13 estudantes tiveram contato com o conteúdo de teste por meio do modelo tradicional e 14 com o modelo *flipped classroom*.

Aprendizado

Os diferentes valores associados a variável aprendizado foram coletados por meio dos testes de conhecimento e a partir disso, foram calculados os valores de $\Delta 1$, $\Delta 2$ e $\Delta 3$. Nesse sentido, $\Delta 1$ caracteriza-se por expor o quanto o aluno aprendeu a partir da instrução realizada por meio dos diferentes modelos pedagógicos. $\Delta 2$ representa o quanto as atividades práticas realizadas após a instrução contribuíram para a aprendizagem efetiva do conteúdo. Por fim, $\Delta 3$ retrata o quanto o aluno aprendeu a partir da iniciativa como um todo.

Para calcular cada um dos valores de Δ, foi necessário estabelecer três fórmulas. Todas elas são baseadas no estudo de Lyra *et al.* (2016) que considera o aprendizado como uma variável experimental durante o seu estudo controlado sobre materiais didáticos.

$$\Delta 1 = Y_{(\mathrm{i})} - X_{(\mathrm{i})}$$

$$\Delta 2 = Z_{(\mathrm{i})} - Y_{(\mathrm{i})}$$

$$\Delta 3 = Z_{(\mathrm{i})} - X_{(\mathrm{i})}$$

em que:

X = Número de acertos no pré-teste;

Y = Número de acertos no teste intermediário;

Z = Número de acertos no pós-teste;

i = {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_\mathrm{n}$}.

A partir da correção dos testes de conhecimento, os dados foram tabulados e os valores para $\Delta 1$, $\Delta 2$ e $\Delta 3$ calculados. Uma análise descritiva foi realizada, disponíveis na Tabela 12. Em relação ao número de acertos no pré-teste, foi possível observar que o grupo controle obteve uma média e uma mediana menor que grupo experimental. No entanto, o desvio padrão amostral revela que a variação do número de acertos foi maior no grupo experimental.

Em geral, os alunos foram bem no pré-teste, conquistando acertos superiores a 50% do número total de questões. Acredita-se que isso tenha ocorrido devido as questões que constituem o pré-teste. Dentre as dez questões, apenas três delas são teóricas. As demais questões foram dedicadas à aplicação prática do critério particionamento em classes de equivalência. Esse critério, por sua vez, é considerado muito simples Paschoal e Souza

(2018a) e pode ser facilmente entendido por aqueles que tem domínio no conteúdo de Teoria dos conjuntos.

Tabela 12 – Número de acertos e medidas de aprendizado para os modelos pedagógicos

	Grupo controle			**Grupo experimental**		
	Média	**Mediana**	**Desvio padrão**	**Média**	**Mediana**	**Desvio padrão**
Número de acertos no Pré-teste	5,15	5,00	0,99	5,86	6,00	1,10
Número de acertos no Teste intermediário	6,46	6,00	0,78	6,50	7,00	0,94
Número de acertos no Pós-teste	6,54	6,00	2,15	8,07	9,00	2,20
$\Delta 1$	1,31	1,00	1,03	0,64	1,00	1,22
$\Delta 2$	0,08	0,00	2,29	1,57	2,50	2,06
$\Delta 3$	1,38	1,00	2,33	2,21	3,00	2,01

Fonte: Elaborada pelo autor.

A análise acerca do número de acertos no teste intermediário revela que a média dos alunos que estudou com *flipped classroom* foi muito próxima a média de acertos dos alunos que estudou com o modelo tradicional. A mediana revelou que 50% do número de acertos dos estudantes do grupo experimental foram superiores a 7. O desvio padrão amostral revela que o número de acertos no teste intermediário referente ao grupo controle teve menos variação que o grupo experimental. Isso pode indicar que os alunos do grupo experimental podem ter conquistado o número de acertos mais diversificado, isto é, enquanto um aluno obteve muitos acertos no teste, outro aluno pode não ter obtido muito êxito no teste.

Os resultados referentes ao número de acertos no pós-teste revelam que o grupo experimental obteve melhores resultados que o grupo controle. A média de acertos do grupo experimental foi de 8,07, enquanto que a média obtida pelo grupo controle foi de 6,54. A mediana também foi verificada e revelou que 50% do número de acertos do grupo experimental foram superiores a 9. Em contraponto, a mediana do grupo controle foi 3 pontos inferior. Esses dados revelam que pelo menos 50% dos estudantes do grupo experimental acertou três questões a mais que os estudantes do grupo controle. Para esse teste, os valores de desvio padrão foram similares.

Buscando entender o impacto do modelo no aprendizado do aluno, inicialmente foi analisada a influência do modelo no aprendizado imediato. Como resultado, observou-se que o grupo de alunos que estudou com o modelo tradicional obteve resultados melhores no aprendizado que o grupo experimental, uma vez que a média de $\Delta 1$ foi equivalente a 1,31 para o grupo controle, enquanto que o aprendizado no grupo experimental fechou com

uma a média de 0,64. Com base nisso, acredita-se que o modelo tradicional impulsionou um aprendizado imediato maior.

Em relação ao $\Delta 2$ foi possível observar que as atividades práticas não impactaram na melhoria de aprendizado no grupo controle, uma vez que o valor médio foi próximo de zero (0,08) e a mediana foi zero. Em contrapartida, essas atividades promoveram o aprendizado dos alunos do grupo experimental, uma vez que a média foi de 1,57 e a mediana alcançou 2,5. Nesse sentido, o número de acertos nos testes de conhecimento, após a atividade prática em cada participante do grupo experimental, aumentou em média 1,57 em relação ao teste intermediário. O valor da mediana revela que mais de 50% dos estudantes desse grupo obtiveram um aumento de 2,5. Nesse sentido, nota-se que as atividades práticas apesar de serem semelhantes as realizadas no grupo controle, favoreceram o modelo *flipped classroom*

Finalmente, $\Delta 3$ foi calculado e analisado, visando entender qual grupo obteve melhores resultados de aprendizado. Nessa análise foi considerado o quanto o aluno aprendeu a partir da iniciativa como um todo (incluindo instrução e atividades práticas). A partir da análise foi possível perceber que a média de $\Delta 3$ para o grupo experimental foi superior a do grupo controle, indicando que em média os alunos do grupo experimental conseguiram aprender mais que os alunos do grupo controle. O valor da mediana do grupo experimental também é superior ao do grupo controle. Ademais, o desvio padrão amostral contribui para essa descoberta, uma vez que revela uma menor variação nos valores de $\Delta 3$ para o grupo experimental.

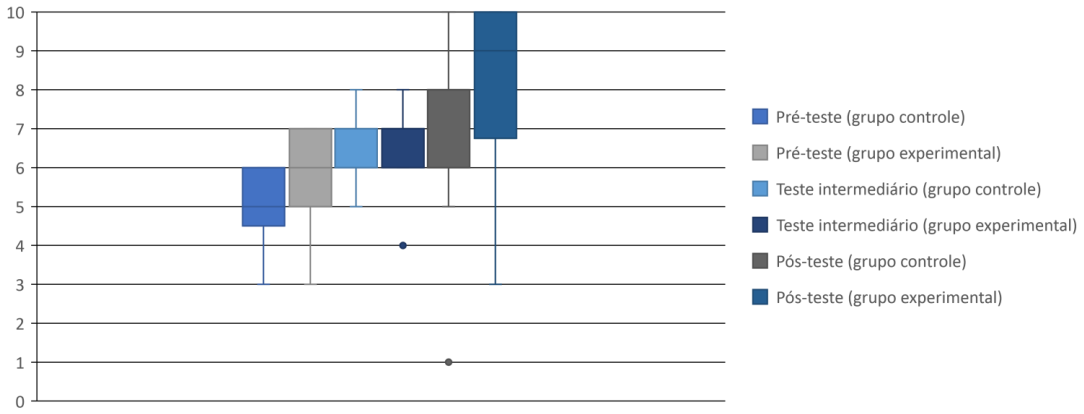
Para finalizar a análise da variável aprendizado, um gráfico do tipo *box plot* foi construído visando apoiar a análise gráfica do número de acertos nos testes de conhecimento, em cada grupo. Conforme é possível observar na Figura 15, na medida em que as iniciativas foram realizadas, o aprendizado aumentou. Esses resultados reforçam o senso de que quanto mais iniciativas forem oferecidas pelo professor, independentemente do modelo, os alunos aprenderão mais. Nota-se que os alunos, independentemente do grupo, obtiveram número de acertos equivalentes após a instrução.

A atividade prática foi essencial para os alunos que estudaram com *flipped classroom*. Ela impulsionou o aprendizado dos alunos, fazendo com que o grupo experimental obtivesse mais acertos no teste de conhecimento que os alunos que estudaram o mesmo conteúdo, que realizaram atividade equivalentes em sala de aula, mas foram instruídos de outra forma (modelo tradicional).

Com os *box plots* foi possível reconhecer valores que são muito diferentes das demais. Durante o teste intermediário do grupo experimental, um aluno obteve um número de acerto muito abaixo dos demais alunos. Similarmente foi revelado que o número de acertos no pós-teste relacionado ao grupo controle contou com um valor distante dos demais. Apesar da existência desses *outliers*, eles não foram removidos na análise descritiva dos

dados.

Figura 15 – Box plot que compara o número de acertos dos alunos nos testes de conhecimento



Fonte: Elaborada pelo autor.

Carga de Trabalho

A segunda variável analisada no decorrer do experimento foi a carga de trabalho, que se refere ao tempo que os alunos "gastaram" para se instruir sobre o conteúdo. Durante a execução do experimento, mais especificamente durante o segundo encontro, os estudantes do grupo experimental tiveram que indicar quanto tempo eles "gastaram" ao se preparar para a aula, assistindo os vídeos. A partir disso, para fazer a análise do tempo gasto de instrução, foi adicionado a cada resposta do indivíduo mais 15 minutos, os quais são providos do tempo utilizado para resolução de dúvidas dos alunos. A Tabela 13 apresenta uma visão geral dos resultados obtidos.

Tabela 13 – Estatística descritiva do tempo gasto pelos alunos do grupo experimental

Grupo	Média	Desvio-Padrão	Mínimo	Máximo	Mediana
Experimental	38 minutos	7,52	22 minutos	50 minutos	40 minutos

Fonte: Elaborada pelo autor.

Para apoiar a análise, foi considerado o tempo médio dedicado para a instrução dos alunos do grupo controle. Esse tempo médio foi de 30 minutos, uma vez que o professor da disciplina fez as suas preleções sobre o conteúdo em exatamente 30 minutos, conforme previsto no planejamento inicial. Nesse intervalo de tempo, o professor apresentou a técnica de teste funcional e detalhou o critério particionamento em classes de equivalência. Ele também solucionou as dúvidas dos alunos na medida que o conteúdo era apresentado.

A partir das respostas dos estudantes do grupo experimental, foi possível observar que o grupo experimental levou em média mais tempo que o grupo controle. O grupo experimental "gastou" um tempo médio equivalente a 38 minutos, enquanto que o tempo médio utilizado pelo grupo controle foi de 30 minutos. O valor mínimo do grupo experimental revela que um aluno possivelmente não assistiu a todos os vídeos, uma vez que levou apenas 22 minutos e o tempo mínimo esperado para esse grupo foi seria 30 minutos e 49 segundos.

O valor máximo, por outro lado, revela que um estudante demorou 50 minutos para estudar o conteúdo, isto é, o aluno dedicou quase uma hora para estudar o conteúdo. A mediana também foi observada nessa variável. O valor da mediana indica que mais de 50% dos estudantes "gastaram" mais de 40 minutos para se preparar para a aula. Com base nisso, presumisse que os estudantes devem ter assistido mais de uma vez os vídeos, pausado para entender melhor o conteúdo, ou assistido novamente a momentos específicos do vídeo.

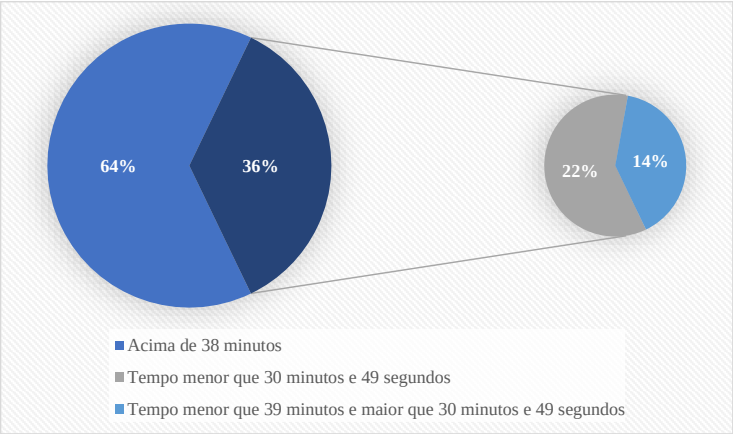
Foi analisada a frequência de participantes que gastaram mais que a média de tempo que o grupo experimental levou ao estudar o conteúdo. Para tanto, o tempo "gasto" por cada estudante do grupo experimental foi classificado em uma de três categorias. A primeira categoria contém os dados dos alunos que gastaram mais de 38 minutos para estudar o conteúdo, a segunda categoria contém os dados dos alunos que levaram menos de 39 minutos e mais de 30 minutos e 49 segundo. A terceira categoria, por sua vez, contém a sequência de dados de tempo que são inferiores a 30 minutos e 49 segundos.

Tomando como base a classificação, foi possível constatar que 64% dos estudantes do grupo experimental levaram mais de 38 minutos para se instruir sobre o conteúdo. Por outro lado, 36% dos estudantes levou menos que o tempo médio gasto pelo grupo. Dentre esses 36%, 22% constituem os estudantes que levaram menos que 30 minutos e 49 segundos. Finalmente, 14% dos alunos levou entre 30 minutos e 49 segundos e 39 minutos. A análise está ilustrada por gráficos na Figura 16.

Por fim, visando compreender se os alunos que estudam com *flipped classroom* "gastam" mais tempo para se instruir sobre o conteúdo que os alunos que estudam esse mesmo conteúdo com o modelo tradicional, o tempo individual que cada aluno do grupo experimental levou foi classificado em dois fragmentos. No primeiro fragmento, foram agrupados os alunos que levaram mais de 30 minutos para estudar o conteúdo. Em contrapartida, no segundo fragmento foram agrupados os alunos que levaram menos de 31 minutos. A Figura 17 contém um gráfico que ilustra os resultados.

Foi possível identificar que 79% dos alunos do grupo experimental levaram mais que o tempo médio "gasto" para a instrução do grupo controle. Apenas 21% dos estudantes levaram menos tempo para se instruir sobre o conteúdo que o grupo controle. Desse modo, apesar de ser uma análise preliminar que considera apenas o tempo de instrução,

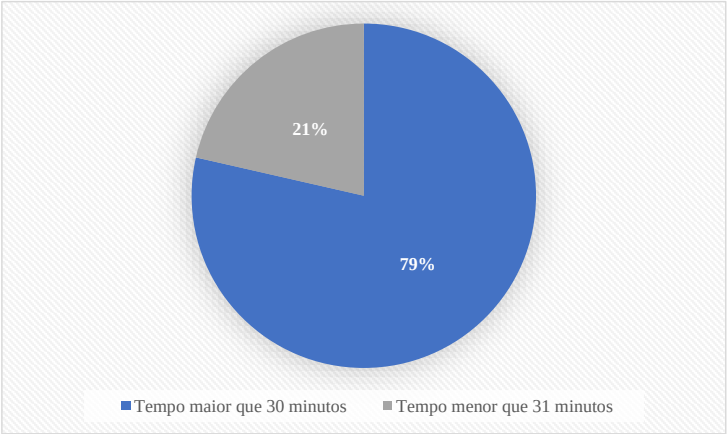
Figura 16 – Relação do tempo gasto pelos estudantes com o tempo médio gasto pelo grupo experimental



Fonte: Elaborada pelo autor.

acredita-se que o aluno que estuda teste de software com o modelo *flipped classroom* leva mais tempo de instrução que o aluno que estuda esse conteúdo com o modelo tradicional de ensino.

Figura 17 – Comparação entre o tempo gasto pelos alunos do grupo experimental com o tempo médio gasto pelos alunos do grupo controle



Fonte: Elaborada pelo autor.

Percepção do Aluno

O último aspecto analisado foi a percepção dos estudantes sobre o modelo*flipped classroom*, após eles terem um contato preliminar com esse modelo. Para tanto, o formulário desenvolvido e utilizado na pesquisa de He *et al.* (2016) foi aplicado. Esse formulário contém catorze questões que são fracionadas em três partes. Na primeira, os alunos devem analisar afirmações e, com base na experiência adquirida, selecionar uma alternativa. Estas alternativas são baseadas em uma escala Likert de seis pontos (1 – discordo total-

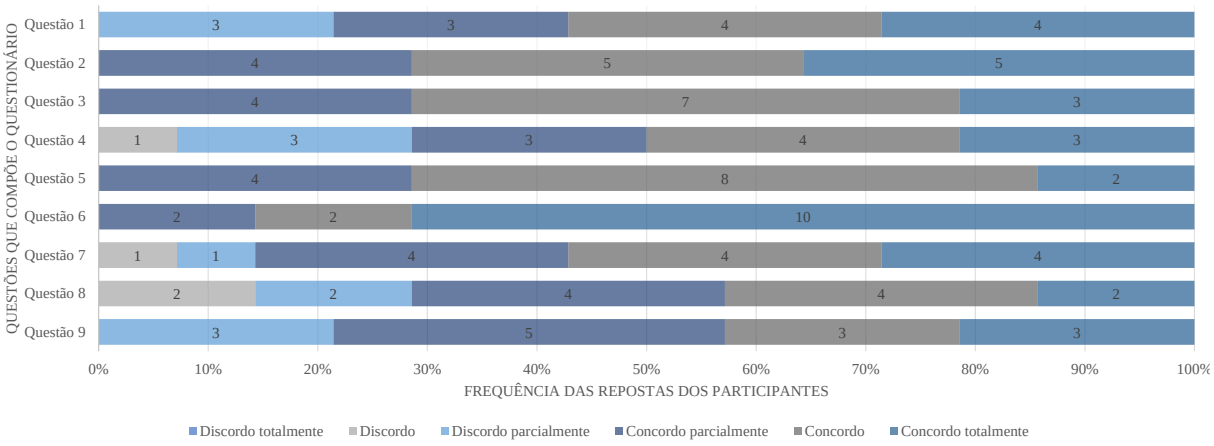
mente; 6 – concordo totalmente). Todas as questões são apresentadas na Tabela 14 e as respostas são ilustradas na Figura 18.

Tabela 14 – As nove primeiras questões que constituem o formulário de percepção

Questão	**Tipo**
(1) Estou muito interessado no conteúdo e na área de teste funcional	Escala Likert (obrigatória)
(2) Este conteúdo ainda será muito útil para mim	Escala Likert (obrigatória)
(3) Estou muito confiante de que vou me sair bem na atividade que eu realizar	Escala Likert (obrigatória)
(4) Vou precisar do conteúdo desta atividade em disciplinas subsequentes	Escala Likert (obrigatória)
(5) Dada a minha situação atual, estou confiante em conseguir uma boa nota na atividade que vou realizar	Escala Likert (obrigatória)
(6) Os vídeos disponibilizados para o estudo foram claros para mim	Escala Likert (obrigatória)
(7) A instrução do professor em sala de aula foi clara para mim	Escala Likert (obrigatória)
(8) Eu prefiro estudar com *Flipped Classroom* em detrimento a um formato de aula tradicional	Escala Likert (obrigatória)
(9) Eu preferiria ter mais aulas de teste de software funcional usando *Flipped Classroom*	Escala Likert (obrigatória)

Fonte: Elaborada pelo autor.

Figura 18 – Respostas dos estudantes sobre suas percepções relacionadas ao modelo *Flipped Classroom*



Fonte: Elaborada pelo autor.

Na primeira questão, os estudantes declaram seu interesse no conteúdo de teste funcional. Como resultado, dentre os catorze estudantes, três (21%) manifestaram que não estavam interessados. Em seguida, os alunos foram convidados a assumir uma posição sobre o conteúdo ser útil para eles ao longo de suas vidas. Obteve-se uma concordância unânime na assertiva, uma vez que nenhum dos entrevistados discordou que o conteúdo ainda será muito útil para eles. No terceiro enunciado, os sujeitos tiveram que declarar se eles estavam muito confiantes que iriam se sair bem nas atividades que iriam realizar após preencher o formulário. Similarmente a segunda questão, todos concordaram estar confiantes.

Durante quarta assertiva, os participantes tiveram que analisar se eles acreditavam que necessitariam do conteúdo em disciplinas subsequentes do curso. Quatro alunos (29%) discordaram, dando a entender que acreditavam que não precisariam do conteúdo. Na sequência, os sujeitos que participaram do estudo declaram sobre a confiança em conquistar uma boa nota na atividade que realizariam após responder o questionário. Para essa assertiva, todos os alunos declarar confiança, uma vez que eles selecionaram somente alternativas que variam de concordo parcialmente até concordo totalmente. Esses estudantes também tiveram que indicar se os vídeos estavam claros. Semelhantemente aos resultados da questão anterior, todas os estudantes assinalaram assertivas que variam entre concordo parcialmente e concordo totalmente.

Na antepenúltima assertiva do primeiro fragmento do formulário, a instrução do professor em sala de aula teve que ser analisada pelos estudantes. Buscou-se averiguar se o momento de resolução de dúvidas havia sido claro para eles. Nessa assertiva, as respostas variaram entre polos de discordo e concordo. Dois alunos (14%) dos catorze declararam insatisfação com o momento de resolução de dúvidas. Em contraponto, os demais participantes (86%) declararam estar satisfeitos.

A oitava questão interrogou os estudantes se eles preferem estudar o conteúdo de teste funcional com o modelo *flipped classroom* ou com o modelo tradicional. Vale ressaltar que essa questão é utilizada na pesquisa de He *et al.* (2016) com o propósito de reconhecer a preferência dos alunos pelo modelo pedagógico. Similarmente, nesse estudo a questão será utilizada como um indicativo de preferencia dos estudantes. 71% dos entrevistados declararam que preferem estudar com o modelo *flipped classroom* e os outros participantes assinalaram alternativas que denotam que eles preferem o modelo tradicional.

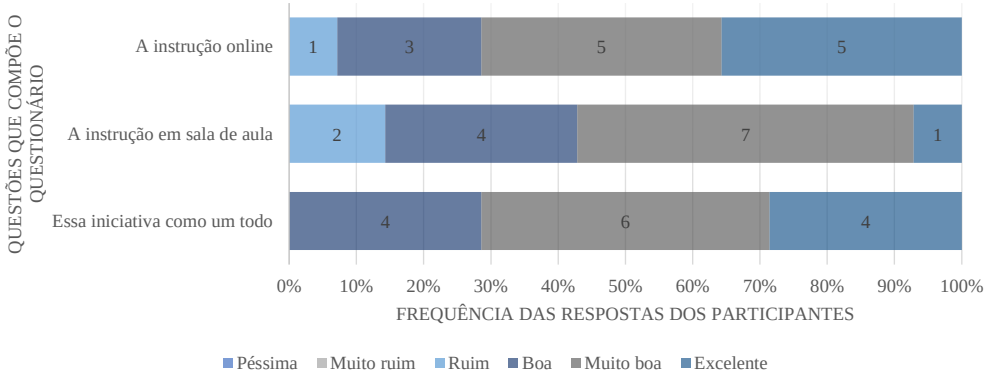
A última afirmativa do primeiro fragmento questionou os participantes se eles preferem mais aulas do conteúdo trabalho com o modelo *flipped classroom*. Como resultado, apenas três estudantes (21%) manifestaram que não possuíam interesse em continuar estudando o conteúdo com *flipped classroom*.

Após os estudantes avaliarem as nove assertivas, com base em suas experiências obtidas com o modelo, no segundo fragmento do formulário, eles foram convidados a

avaliar a qualidade geral de três itens: a instrução *online* (isto é, os vídeos instrucionais), a instrução realizada em sala de aula e a iniciativa como um todo. Como alternativa para avaliar a qualidade, foram oferecidas aos participantes uma escala de seis pontos que varia entre péssima qualidade e excelente qualidade.

Em relação à instrução online, um estudante (7%) declarou ela como um ruim. Os demais declararam a qualidade como boa, muito boa e excelente. Quanto a instrução em sala de aula, dois participantes (14%) alegaram que a qualidade estava ruim. Na sequência, para finalizar a avaliação relacionada à qualidade dos itens, nenhum estudante declarou de forma negativa a iniciativa como um todo. Com base nisso, acredita-se que os estudantes em geral, acharam que a qualidade da iniciativa foi boa. Os resultados desse fragmento são apresentados na Figura 19.

Figura 19 – Avaliação sobre a qualidade geral de itens do modelo *Flipped Classroom*



Fonte: Elaborada pelo autor.

Após doze assertivas com opções fechadas de respostas, as últimas duas questões do formulário foram abertas. Na primeira questão, os alunos foram questionados se eles realmente gostaram do modelo *flipped classroom*. Nessa mesma questão eles precisariam discutir o porquê da resposta emitida. Em geral, dos 14 estudantes apenas um indicou por que ele não gostou. Os demais realizaram algumas declarações. Um estudante mencionou que acredita que com o modelo, ele conseguiu levar menos tempo para aprender o conteúdo. Três estudantes mencionaram a flexibilidade de estudar com o modelo, descrevendo que cada estudante pode aprender em seu próprio ritmo. Outro aluno indicou que o modelo oportunizou o aprendizado do conteúdo sem interrupções. Um participante mencionou que gostou, no entanto, ele sentiu falta de apoio durante o estudo do conteúdo preliminar a aula, especialmente para solucionar dúvidas. Em relação ao aluno que mencionou que não gostou do modelo, ele relatou que esse modelo não consegue substituir uma aula tradicional. Todas as respostas obtidas na questão foram organizadas no Quadro 1.

Por fim, na última questão do formulário, os estudantes foram convidados a apresentar reclamações e sugestões de melhoria. Para essa questão, seis estudantes descreveram

que não tinham reclamações sobre a iniciativa. Os demais realizaram alguns apontamentos interessantes. Um aluno mencionou que os vídeos eram muito longos e que ele não encontrou a opção para acelerar a execução do vídeo. Apesar desse apontamento, é possível acelerar a execução do vídeo com *players* específicos. Em decorrência dessa resposta, como lição aprendida, têm-se o fato de recomendar um *player* para oportunizar a configuração de velocidade de execução.

Um outro aluno mencionou que um problema que poderia ter surgindo e não foi previsto pelo professor ao adotar o modelo é que os alunos poderiam ter dúvidas ao estudar o conteúdo. De fato, é um problema que ainda não está sendo solucionado pela pesquisa e prática de *flipped classroom*, mas que será abordado no Capítulo 6. Em razão do estudo teórico do modelo ser realizado "antes da aula", um aluno mencionou que o tempo de aula presencial poderia ser minimizado. As demais sugestões emitidas pelos alunos estão disponíveis no Quadro 2.

Quadro 1 – Percepções dos estudantes que estudaram com *Flipped Classroom*

	Você gostou ou não do modelo *Flipped Classroom*? Por que?
1	Gostei, pois na atualidade, o aluno pode encontrar mais facilmente as informações
2	Gostei do modelo, pois conseguiu passar o conteúdo sem interrupções e corretamente
3	Eu gostei, porque levei menos tempo para aprender o conteúdo
4	Gostei, pois permite que cada aluna aprenda o conteúdo em seu próprio ritmo, desde que o material fornecido seja de boa qualidade
5	Além da instrução do professor presencialmente, é possível atacar os conteúdos de maneira individual, o que flexibiliza e amplia o processo de aprendizado, portanto gostei .
6	Gostei, o conteúdo é bem didático, mostra de maneira clara.
7	Gostei do modelo pois o conteúdo foi bem objetivo e pude entendê-lo no meu ritmo, conseguindo facilmente acompanhar a revisão em aula
8	Gostei, no entanto o conteúdo estudado pareceu relativamente simples, o que contribuiu para a facilidade no aprendizado. Para conteúdos mais complexos poderia gerar mais dúvidas
9	Gostei, pois oferece ao aluno um embasamento prévio do tema da aula
10	Gostei, pois assisti o vídeo no momento que gostaria de aprender e não estava preocupado em realizar atividades de outras disciplinas
11	Gostei, acho que dessa forma o aprendizado é mais eficiente visto que cada um pode estudar no seu próprio ritmo
12	Gostei
13	Achei interessante, porém durante o estudo senti muita falta da possibilidade de perguntar a respeito do conteúdo
14	Acredito que é mais adequado como um complemento a aula e não substituir a aula tradicional

Fonte: Elaborada pelo autor.

Quadro 2 – Reclamações e sugestões de melhoria para o modelo *Flipped Classroom*

	Qual é a sua principal reclamação sobre esta experiência e quais são suas recomendações para nós fazermos melhorias?
1	Os vídeos era muitas vezes prolixos e não havia a opção de acelerar a execução
2	Um possível problema, que não ocorreu, seria a necessidade de tirar dúvidas durante o estudo
3	Creio que o tempo de aula presencial deveria ser reduzido
4	É necessário a dedicação de um tempo extra aula para o estudo, o que dificulta o estudo de outros conteúdos.
5	Acredito que a revisão dos conceitos a partir de exemplo que foi dado no vídeo (exatamente como foi explicado) seria mais efetivo para sanar eventuais dúvidas
6	Senti fala de praticar o conteúdo durante o estudo. Talvez um teste múltipla-escolha seria interessante.
7	Acho que eu preferiria estudar o conteúdo em forma de texto do que em vídeo
8	Não tenho reclamações
9	Não tenho reclamações
10	Exemplos que pode elucidar o conteúdo ou gerar melhor conhecimento do conteúdo
11	Não tenho reclamações
12	Não tenho reclamações
13	Não tenho reclamações
14	Não tenho reclamações

Fonte: Elaborada pelo autor.

3.3.5 Síntese das Análises

Este experimento foi planejado com o propósito de identificar o impacto do modelo *flipped classroom* no aprendizado, na carga de trabalho e na percepção do estudante de teste de software. Para mensurar o aprendizado, testes de conhecimentos foram planejados e aplicados durante o desenvolvimento do experimento. A carga de trabalho, por sua vez, foi mensurada por meio do tempo médio gasto pelo grupo controle e o tempo que cada aluno do grupo experimental levou ao estudar o conteúdo. Por outro lado, as percepções dos estudantes foram coletadas por intermédio de um formulário que foi previamente desenvolvido por He *et al.* (2016) e adaptado para o contexto desta pesquisa.

A partir das respostas dos estudantes nos testes de conhecimento, foi possível calcular o quanto cada aluno aprendeu após a instrução ($\Delta 1$), o quanto as atividades ativas influenciaram no aprendizado ($\Delta 2$) e quanto de conhecimento os alunos conseguiram reter ao final das iniciativas ($\Delta 3$). Com base nisso, foi possível reconhecer que o modelo tradicional de ensino, em um primeiro momento, impulsionou melhores resultados de aprendizado no estudante. No entanto, a partir da realização de atividades prática, o grupo que estudou com *flipped classroom* superou o grupo que estudou com o modelo tradicional. A realização da atividade impulsionou os valores de $\Delta 3$, oferecendo indícios que os aluno que aprendem teste funcional com *flipped classroom* conseguem aprender mais que os alunos que estudam com o modelo tradicional.

Em relação à carga de trabalho, foi possível observar que os alunos que estudam

com *flipped classroom* levam mais tempo que os alunos que estudam esse mesmo conteúdo com o modelo tradicional. Contudo, é importante reforçar que durante esse experimento foram considerados apenas dois momentos previsto pelo modelo *flipped classroom*, mais especificamente relacionados ao tempo de instrução. Nesse sentido, o tempo total gasto pelos grupos não foi mensurado. Acredita-se que para uma análise mais profunda sobre a temática, é importante considerar todos os momentos, inclusive o tempo gasto pelos alunos "depois da aula", isto é, verificar o tempo gasto nas iniciativas como um todo.

Quanto às percepções dos estudantes sobre o modelo *flipped classroom*, foi possível contatar que de modo geral, eles gostaram desse modelo pedagógico. A maioria dos estudantes mencionou ao relatar sobre sua experiência que prefere estudar teste funcional com esse modelo em comparação com o tradicional.

A partir dos resultados foi possível responder as três questões de pesquisa que permeiam o experimento. A resposta para cada uma delas é apresentada na Tabela 15. Salienta-se que as respostas foram construídas com bases nos resultados que, por sua vez, basearam-se em análises descritivas. Assim, para responder as perguntas, as hipóteses não foram testadas com métodos de estatística inferencial.

Tabela 15 – Resumo das respostas às questões de pesquisa do segundo experimento

Questão de pesquisa	**Resposta**
O aprendizado de teste de software por meio do modelo *Flipped Classroom* é maior do que o aprendizado por meio do modelo de ensino tradicional?	Sim. O aprendizado foi maior nos estudantes que estudaram com *Flipped Classroom.*
Os alunos de teste de software que estudam com *Flipped Classroom* passam mais tempo estudando do que os alunos que estudam com o modelo tradicional?	Sim. A média de tempo de instrução gasto pelos alunos que estudaram com *Flipped Classroom* é maior que o tempo médio gasto pelos alunos do modelo tradicional.
Os alunos de teste de software que estudam com *Flipped Classroom* preferem esse modelo pedagógico em detrimento ao modelo tradicional?	Sim. Os alunos que tiveram uma experiência com o modelo *Flipped Classroom* preferem esse modelo ao estudar o conteúdo.

Fonte: Elaborada pelo autor.

3.3.6 Ameaças à Validade

Com base no planejamento, execução e análise de resultados, as ameaças relacionadas a este experimento são descritas a seguir.

Validade de conclusão: Os resultados obtidos em um estudo experimental podem sofrer influência de fatores, como a confiabilidade das medidas, testes estatísticos, o ambientes de operação e situações aleatórias ocorridas no ambiente de execução.

Nesse sentido, para mitigar ameaças associadas a esses fatores, algumas atitudes foram tomadas. Em relação a confiabilidade das medidas, o aprendizado foi mensurado por meio de testes de conhecimento que avaliam o aspecto de memória do estudante, conforme as definições de Mayer (2002). Esses testes contêm problemas que precisam ser solucionados pelos alunos. Essa característica dos testes foi atribuída, pois de acordo com Mayer (2002) o aprendizado ocorre quando o estudante é capaz de aplicar o conhecimento adquirido em uma resolução de problema.

Como estratégia para mensurar a carga de trabalho, a medida utilizada no estudo experimental foi definida tomando como base o estudo experimental conduzido por He *et al.* (2016). Nesse sentido, foi estabelecido que os alunos do grupo experimental precisariam anotar o tempo que cada um levou para estudar (em minutos) e, na sequência, esse tempo sofreria um acrescimento a partir de um somatório com o tempo utilizado pelos alunos para a resolução de dúvidas. Quanto a medida para mensurar a preferência do estudante em estudar com *flipped classroom*, ela também foi definida com base no experimento de He *et al.* (2016).

Em relação aos testes estatísticos, inicialmente, haviam sido planejados teste semelhantes aos utilizados no primeiro experimento. No entanto, a análise inferencial não foi realizada após a coleta dos dados, uma vez que não foi possível aproveitar todos os dados coletados. Adicionalmente, alguns alunos optaram por não disponibilizar os seus dados para análise no experimento. Nesse sentido, apenas análises descritivas foram realizadas, as quais apresentam apenas um direcionamento tomando como base as medidas de posição e medidas de dispersão.

Por fim, para mitigar as situações aleatórias que poderiam ocorrer no ambiente de execução do experimento algumas ações semelhantes às tomadas no primeiro experimento foram definidas. Para minimizar barulhos provocados no ambiente externo a sala de aula (*e.g.* corredor, escadas), foram selecionadas salas de aula de uma extensão do campus, o qual não possui transição de alunos no turno e dias da semana em que o experimento foi executado. Os estudantes da disciplina também foram notificados que eles deveriam tomar cuidado para não gerar tumulto ao adentrar na sala de aula em que o experimento seria executado.

Validade interna: As mudanças que estão sendo observadas nas variáveis de saída devem ser atribuídas ao tratamento e não a outros fatores. Nesse sentido, é importante ponderar sobre possíveis fatores que podem interferir na relação causa-efeito. Fatores que podem ter oferecido ameaças à validade interna do estudo foram ponderados durante o planejamento do estudo, são eles: qualidade do material didático, aula ministrada pela professora ao grupo controle, desistência dos participantes durante a execução do experimento.

O material didático selecionado para os alunos do grupo experimental utilizar, para se instruir sobre o conteúdo, pode influenciar nos resultados do estudo, os quais refletem em variação das variáveis de saída. Nesse sentido, são elementos que precisaram ser definidos com cuidado. Para tanto, eles foram selecionados dentre um conjunto de outros materiais por quatro pesquisadores. O pesquisador principal desse estudo selecionou alguns materiais, a orientadora da pesquisa que possui mais de 15 anos de experiência na área revisou o material (vídeos) em conjunto com um doutorando com mais de 4 anos de experiência na área de engenharia de software e a professora da disciplina. Essa última, com mais de 10 anos de experiência na área.

Durante a seleção do material, foram considerados alguns requisitos básicos que deveriam ser contemplados para serem estabelecidos e disponibilizado aos alunos. Foi definido que o material deveria ser em formato de vídeo, visando facilitar o ensino, uma vez que por meio dos vídeos os estudantes poderiam assistir quantas vezes eles precisam, poderiam repetir apenas momentos específicos, executar na velocidade que eles quisessem. O vídeo teria que ser no idioma nativo dos participantes, com tempo de duração inferior a 20 minutos ao total, uma vez que Bergmann e Sams (2012) recomendam que os vídeos não podem ocupar muito tempo dos estudantes, senão eles desistirão de assisti-los.

A aula ministrada pela professora da disciplina ao grupo de alunos que estudou teste funcional e particionamento em classes de equivalência foi outro elemento que precisou ser considerado. Um profissional sem muito experiência com a área, por exemplo, pode prejudicar o aprendizado do grupo controle. Nesse sentido, visando mitigar uma possível ameaça, a professora da disciplina com mais de 10 anos de experiência na área, com pesquisa recente sobre teste de software ministrou a aula para o grupo controle. A aula foi planejada para abordar o mesmo conteúdo que o grupo experimental.

Outro problema que pode oferecer ameaça à validade do experimento é a desistência dos estudantes durante a condução/execução do experimento. Para mitigar essa ameaça, o experimento foi considerado como uma atividade natural da disciplina. Desse modo, os alunos foram notificados que uma atividade diferente seria executada na disciplina, que não contabilizaria qualquer tipo de nota, mas seria uma forma diferenciada de ensinar o conteúdo de teste funcional.

Validade de constructo: o design do experimento pode oferecer ameaça à validade do experimento. Em relação aos instrumentos, foram definidos e utilizados apenas um método para mensurar cada uma das variáveis (aprendizado, carga de trabalho, percepção do aluno). No caso do aprendizado, foi utilizado somente testes de conhecimento. No entanto, há relatos na literatura que os testes não são formas exclusivas para mensurar o conhecimento real dos estudantes. Outras formas como realização

de problemas, apresentação, listas de exercícios, autoria de material podem ser utilizados. Apesar disso, os testes de conhecimento foram utilizados porque são mais simples de serem aplicados e o tempo disponível na disciplina para execução da disciplina possuía um limite.

A aplicação dos testes de conhecimento também pode produzir ameaça à validade, uma vez que os estudantes podem memorizar as questões e alternativas do teste aplicado. Para minimizar essa ameaça, foram realizadas pequenas modificações nos enunciados das questões, alteradas as respostas e a ordem das mesmas, em cada uma das variações do teste (pré-teste, teste intermediário e pós-teste). As variações buscaram não comprometer a estrutura da questão e/ou alternativas. Essa ação foi semelhante a realizada no primeiro experimento, quando foram utilizados testes para mensurar o desempenho acadêmico.

Em relação à métrica utilizada para mensurar a carga de trabalho, o tempo de instrução foi considerado. Apesar do tempo de instrução não refletir toda a carga de trabalho do estudante, uma vez que não considera o tempo que os alunos "gastam" para realizar atividades, listas de exercícios ou tarefas após a aula, acredita-se que a métrica consegue refletir um período de tempo fiel. De acordo com He *et al.* (2016), mensurar o tempo não é uma atividade trivial, contudo iniciativas precisam ser feitas. Desse modo, como estratégia para conseguir ter uma visão sobre a variável "carga de trabalho", o tempo de instrução foi observado.

Para ter uma visão geral sobre a percepção do aluno sobre *flipped classroom* foi utilizado apenas uma adaptação do questionário de He *et al.* (2016). Apesar de ser um único instrumento, ele já foi utilizado em outros estudos experimentais e possui pelo menos uma avaliação. Acredita-se que usar um formulário já utilizado e reconhecido pela comunidade é uma melhor opção do que construir outro instrumento, com propósito igual.

Validade externa: Uma ameaça que pode estar presente em estudos experimentais é planejar e executar um experimento e não conseguir generalizar os resultados para o contexto o qual o objeto de investigação e as respectivas variáveis fazem parte. Em especial, acredita-se que essa ameaça foi minimizada neste experimento por meio da seleção correta dos sujeitos e do contexto. O experimento foi executado com estudantes de engenharia de software de um curso de graduação em Computação. De acordo com Paschoal e Souza (2018b), em geral, na ausência de uma disciplina específica sobre teste de software, os critérios e as técnicas de teste de software são ensinados em disciplinas de engenharia de software. Adicionalmente, a técnica e o critério abordado são tópicos amplamente abordados nessa área Paschoal e Souza (2018b), Paschoal e Souza (2018a), facilitando a sua execução em cenários reais de ensino.

3.3.7 Apresentação e Empacotamento

Após a definição, o planejamento, a execução e a análise dos resultados, a última etapa do experimento é a apresentação e o empacotamento dos materiais e dados coletados. Diferentemente do primeiro experimento, este experimento ainda não foi submetido para um comitê de um evento ou de um *journal* especializado. Assim, ele ainda não está publicado em um veiculo de divulgação de conhecimento científico. Espera-se divulgar os resultados em veículo especializado da área, visando ampliar a visibilidade do estudo.

Todo o material preparado e utilizado no decorrer do experimento, como materiais didáticos, testes de conhecimento, formulários, dentre outros, foram publicados em um ambiente de acesso público. Os dados coletados no experimento e as análises também foram disponibilizadas, visando facilitar o acesso ao material por parte da comunidade acadêmica que tem interesse em investigar o modelo pedagógico *flipped classroom*. Além disso, a disponibilizado de todo o material em um pacote de laboratório é importante para oportunizar a replicação do experimento. O pacote de laboratório está disponível em: <http://bit.ly/2FSw5qj>.

3.4 Terceiro Experimento

Tanto no primeiro experimento quanto no segundo experimento, os estudantes de teste de software realizaram tarefas que são previstas para serem realizadas por profissionais na indústria do software, são elas: reconhecer os elementos requeridos para satisfazer os critérios adotados e, com base no reconhecimento, definir casos de testes que consigam cobrir o software ou o programa em teste. Apesar dessas tarefas serem ensinadas aos alunos durante o experimento, em nenhum momento elas foram planejadas para serem analisadas.

Nesse sentido, após reconhecer o impacto do modelo *flipped classroom* no desempenho acadêmico, na motivação, no aprendizado, na carga de trabalho e na percepção dos estudantes de teste de software, notou-se a importância de dar atenção as tarefas específicas do teste de software. Assim, foi definido um terceiro experimento, que retoma a proposta da atividade de teste de software, tendo o princípio de reconhecer se os alunos que estudam com o modelo *flipped classroom* conseguem identificar, com base na especificação de um projeto, corretamente os requisitos de teste (elementos requeridos) para o teste funcional

As outras tarefas do teste de software não são consideradas neste experimento, uma vez que a escrita de casos de teste é muito específica do aluno. Alguns estudantes podem criar casos de testes que conseguem cobrir mais de um elemento requerido, criando um conjunto de casos de teste otimizado. Por outro lado, alguns alunos podem criar casos de teste para cada elemento requerido (*i.e.*, um conjunto de casos de teste com menor

qualidade). Por esse motivo, o experimento não considera a implementação de casos de teste.

3.4.1 Definição

Objetivo

Este estudo experimental tem o objetivo de **analisar** o modelo *flipped classroom*, **com o propósito de** verificar seu impacto quando utilizado para ensinar estudantes a reconhecer os elementos requeridos de um software em teste para satisfazer um determinado critério de teste, **com respeito** à eficiência, eficácia e efetividade, **do ponto de vista** do pesquisador, **no contexto** estudantes de Engenheira de Software, estudando o conteúdo de teste de software.

Objeto

O objeto de estudo deste experimento é o modelo pedagógico *flipped classroom* instanciado em uma iniciativa para ensinar teste de software. A iniciativa é baseada nas descrições sobre planejamento e implementação de *flipped classroom* de Bergmann e Sams (2012).

Foco Qualitativo

O foco qualitativo é a mensuração da efetividade do aluno que aprende, por meio de *flipped classroom*, a identificar os elementos requeridos para satisfazer o critério particionamento em classes de equivalência.

Perspectiva

Este estudo tem uma perspectiva de realização em um ambiente acadêmico, isto é, em uma disciplina de um curso de graduação em Computação, reconhecido pelo Ministério da Educação. O estudo foi planejado e executado por um acadêmico de pós-graduação em Computação, com orientação da professora que ministra a disciplina em que o estudo foi idealizado e executado.

Contexto

Este estudo experimental foi conduzido no primeiro semestre de 2018, com alunos do 5º semestre do curso de bacharelado em ciências de computação, oferecido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo. Os alunos estavam devidamente matriculados na disciplina de engenharia de software. O experimento foi executado no mês de abril, na sequência do conteúdo introdutório de teste

de software (*i.e.*, definição de teste de software, conceitos e terminologias, uma visão geral sobre as técnicas e fontes de informação utilizadas para condução da atividade).

3.4.2 Planejamento

Seleção do Contexto

Este estudo experimental prevê um processo *offline*, em que os estudantes são entrevistados em momentos específicos durante a execução do estudo. Os participantes utilizados no experimento são estudantes, de uma disciplina de engenharia de software de um curso graduação, em que o conteúdo de teste de software é ensinado como uma unidade da disciplina.

Em relação à realidade do experimento, ela pode ser considerada real, uma vez que o conteúdo e as atividades foram planejadas para execução em ambiente real de ensino. Por consequência do experimento englobar uma comparação entre dois modelos pedagógicos (*flipped classroom* versus modelo tradicional), o impacto do modelo tem um caráter específico.

Definição das Hipóteses

O terceiro experimento também tem o intuito de subsidiar o estabelecimento de evidências para responder a primeira questão de pesquisa "É viável ensinar conteúdos de teste de software com o modelo pedagógico flipped classroom?". Para contribuir, este experimento em particular aborda uma questão de pesquisa específica, ainda não investigada na literatura da área: "Os alunos que estudam com *flipped classroom* tem mais efetividade ao reconhecer os elementos requeridos de um software em teste que os alunos que estudam com o modelo tradicional de ensino?". Buscando responder essa questão, as seguintes hipóteses foram formalizadas:

- **Hipótese Nula:** Não existe diferença em termos de efetividade em relação à identificação dos elementos requeridos feita por alunos que estudam com *flipped classroom* e por alunos que estudam com o modelo tradicional.

  H_0: Efetividade $_{\text{Flipped Classroom}}$ = Efetividade $_{\text{Tradicional}}$

- **Hipótese Alternativa:** Há uma diferença na efetividade em relação a identificação dos elementos requeridos feita por alunos que estudam com *flipped classroom* e por alunos que estudam com o modelo tradicional.

  H_1: Efetividade $_{\text{Flipped Classroom}}$ $\neq$Efetividade $_{\text{Tradicional}}$

Seleção dos Sujeitos

Os estudantes escolhidos para participar deste experimento foram selecionados por conveniência. 93 alunos da disciplina de engenharia de software do curso de Ciências de Computação foram convidados para participar do experimento. Esses alunos são provenientes de duas turmas, uma vez que no curso de Bacharelado em Ciências de Computação, anualmente há o ingresso de aproximadamente 100 estudantes, formando duas turmas.

Diante da escolha dos sujeitos, ficou definido que uma turma de alunos teria a experiência de estudar o critério de particionamento em classes de equivalência com o modelo *flipped classroom* enquanto que outra estudaria o mesmo conteúdo com o modelo tradicional. Assim, iniciativas de ensino baseadas nesses modelos foram definidas, as quais serão descritas com detalhes na Seção 3.4.3.

Seleção das Variáveis

O presente experimento investiga os valores de uma variável em particular, a efetividade. Para compreender os valores da variável, algumas ações foram tomadas visando controlar os fatores que influenciam direta ou indiretamente essa variável. A seguir são descritas as variáveis que representam as influências que podem afetar o resultado do experimento (variáveis dependentes) e as variáveis que acomodam os valores dos resultados do experimento (variáveis independentes).

Variáveis Dependentes

A efetividade é uma variável que é definida por Cury (2006) como o alcance de eficiência e eficácia ao mesmo tempo pelo sujeito, durante a realização de uma determinada atividade. Assim, para conseguir reconhecer os valores assumidos pela variável efetividade ao final do experimento, as variáveis eficiência e eficácia foram consideradas:

- **Eficiência:** A eficiência remete ao tempo que os participantes consumiram para realizar a atividade proposta. Em outras palavras, o tempo "gasto" pelos estudantes durante o reconhecimento dos elementos requeridos para satisfazer o critério de particionamento em classes de equivalência.

- **Eficácia:** A eficácia remete ao número de elementos requeridos que satisfazem o critério particionamento em classes de equivalência para o projeto de software definido, identificado pelos participantes. Um elemento só é contabilizado como correto se o participante o identificou de forma correta e adequada. Para verificar se o aluno identificou os elementos corretos e de forma adequada, um oráculo de teste foi previamente definido.

- **Efetividade:** A efetividade é uma combinação dos resultados para eficiência e eficácia.

Variáveis Independentes

Neste experimento as seguintes variáveis de entrada foram consideradas:

- **Modelos pedagógico:** consiste nos modelos pedagógicos utilizados para instruir os estudantes durante o ensino do critério de particionamento em classes de equivalência. Dois tratamento são explorados neste experimento, a saber:

  Tratamento A: os alunos são instruídos por meio do modelo *flipped classroom*;

  Tratamento B: os alunos são instruídos por meio do modelo tradicional de ensino.

- **Especificação do software:** consiste na especificação do programa que será utilizado pelos participantes durante a condução do experimento. É a partir do projeto que os alunos devem reconhecer os elementos requeridos para satisfazer o critério particionamento em classes de equivalência.

- **Identificação dos elementos requeridos:** consiste na atividade que deve ser realizada pelos participantes durante a execução do experimento.

Salienta-se que outras variáveis como o conteúdo e a experiência dos participantes também foram destacadas como variáveis independentes deste experimento. Uma descrição sobre cada uma dessas variáveis foi apresentada no primeiro experimento, na Seção 3.2.2.

Descrição da Instrumentação

Para apoiar a execução do estudo experimental, alguns materiais foram produzidos e outros foram recuperados dos experimentos anteriores. Os materiais didáticos utilizados tanto para instruir os estudantes do grupo controle, quanto os estudantes do grupo experimental foram reaproveitados do segundo experimento, uma vez que eles haviam sido selecionados de maneira criteriosa e foram apreciados positivamente pelos alunos.

Para a aula tradicional, foi reutilizado o material didático selecionado e usado durante as aulas tradicionais do primeiro experimento. O ambiente virtual de aprendizagem Moodle, por sua vez, também foi usado neste experimento para apoiar a publicação dos materiais de estudo aos alunos do grupo experimental. Similarmente, grupos foram criados dentro do ambiente, visando evitar que alunos que não deveriam ter acesso ao material conseguissem acesso ao mesmo.

O formulário de caracterização dos sujeitos desenvolvido para o primeiro estudo experimental também foi reaproveitado, uma vez que neste experimento foi definido que seriam coletados os dados sobre o conhecimento (ou a experiência) que os participantes possuíam previamente na área de teste de software. Como neste experimento não havia motivo para aplicar testes de conhecimento, dado que a sua intenção não foi mensurar o conhecimento do aluno, o formulário de caracterização se tornou mais adequado para este experimento.

Especialmente, foi definido uma especificação de uma função de um software que, posteriormente, seria utilizada pelos alunos durante uma atividade em que eles deveriam reconhecer os elementos requeridos necessários para satisfazer o critério particionamento em classes de equivalência. Para tanto, foi definida uma especificação sobre a função denominada *NextDate*. Essa função retorna uma data (*dia/mês/ano*), com base em uma data de entrada fornecida pelo usuário.

A função foi selecionada pelo investigador principal (autor deste trabalho) em conjunto com sua orientadoraApós a seleção, o pesquisador e a professora definiram um oráculo que contém o número mínimo de elementos requeridos que precisavam ser identificados pelos estudantes.

A atividade dos alunos foi estipulada a partir da definição da especificação que os alunos teriam que analisar. Inicialmente, os alunos precisariam anotar o tempo gasto em cada uma das etapas (leitura da especificação, reconhecimento de elementos requeridos/ classes de equivalência). A atividade consistiu na leitura da especificação e na definição das classes de equivalência para a especificação.

Por fim, um termo de consentimentos foi desenvolvido, visando esclarecer os sujeitos da pesquisa qual é o objetivo do estudo e como o mesmo impacta na sociedade. O termo contém informações sobre os deveres dos participantes e as obrigações dos pesquisadores.

3.4.3 Operação

Preparação

A primeira ação tomada pelo pesquisador para preparar a execução do experimento foi planejar com a professora da disciplina o número de aulas necessário para a realização do estudo experimental. Nesse sentido, foi estipulado que o experimento seria executado como uma atividade regular da disciplina. Assim, todos os estudantes participariam do estudo, mas optariam por disponibilizar ou não os dados para a análise. Uma ação similar havia sido utilizada no segundo experimento, uma vez que o estudo não poderia atrapalhar o andamento da disciplina e não haveria outras oportunidades na disciplina para o conteúdo ser retomado.

Como a disciplina é ministrada para duas turmas distintas, foi definido que uma

turma formada por 42 alunos seria ensinada pelo modelo tradicional de ensino, enquanto que a outra turma constituída por 51 alunos seria ensinada pelo modelo *flipped classroom*. Essa definição, ocasionou no pré-estabelecimento do número de aulas necessário para cada turma. Para a turma que seria ensinada pelo modelo tradicional, foi previsto uma aula para o ensino do critério e outra aula para a condução da atividade.

Em contrapartida, para o ensino do critério com o modelo *flipped classroom* foi prevista um momento em uma aula que antecede a aula de resolução de dúvidas, para explicar aos alunos que alguns materiais didáticos (vídeos instrucionais) haviam sido disponibilizados e que eles deveriam estudar o conteúdo, se preparando para uma aula que seria constituída por resolução de dúvidas e de atividades. Nesse modelo, também foi previstas duas aulas. Uma para a resolução de dúvidas e de atividades de aplicação do critério e outra para a realização da atividade.

Após a definição da instrumentação, todos os materiais foram preparados e revisados. Algumas modificações foram feitas no formulário de caracterização do sujeito. O termo de consentimento foi preparado para ser aplicado, assim como o ambiente virtual de aprendizagem Moodle. Nesse caso, como haviam duas turmas, no ambiente Moodle foram registrados dois cursos. Os alunos de cada turma não interagiam com os da outra turma. Assim, cada turma possuía um espaço para acesso ao material e os alunos de uma turma não conseguiam visualizar o material da outra outra.

Foi preparada uma aula tradicional para os estudantes do grupo controle, em que o professor ministrou o conteúdo durante todo o período da aula (aproximadamente 110 minutos). Nessa aula, os alunos assistiram as preleções feitas pelo professor, solucionariam suas dúvidas sobre as preleções, e teriam um tempo para resolver alguns exercícios simples que foram disponibilizados pelo professor no final do conjunto de slides. Esses exercícios poderiam ser finalizados "depois da aula".

Também foi preparada uma aula com o modelo *flipped classroom*, baseando-se nas recomendações para implementação do modelo descritas por Bergmann e Sams (2012). Durante a aula, os estudantes teriam um tempo para solucionar as suas dúvidas e realizar alguns exercícios de aplicação do critério, com a supervisão da professor e de um estudante de pós-graduação que realizava estágio de docência na disciplina.

Execução

Conforme descrito na seção anterior, este experimento foi realizado em duas turmas. A seguir serão apresentadas todas atividades realizadas em cada uma das turmas, durantes todo o experimento

Turma A –- Grupo Experimental: O grupo experimental executou atividades em quatro estágios distintos:

Estágio 1: Ao final de uma aula tradicional da disciplina de engenharia de software, em que conceitos e terminologias de teste de software haviam sido ensinado, a professora da disciplina explicou que adotaria uma abordagem diferente ao ensinar o critério de particionamento em classes de equivalência na disciplina. Ela explicou que utilizaria um modelo pedagógico que oportunizava o momento disponível em sala de aula para resolução de dúvidas sobre o conteúdo teórico, acompanhamento de atividades dos alunos e resolução de dúvidas que eventualmente surgem durante a realização da atividade. Para tanto, os estudantes teriam que estudar preliminarmente o conteúdo antes da aula.

Nesse estágio, antes de liberar os alunos da aula tradicional, eles receberam orientações para acessar o material didático disponível no Moodle. Também foi solicitado aos alunos que preenchessem um formulário. Esse formulário tem o propósito de compreender se os estudantes já haviam estudado em outro momento o critério particionamento em classes de equivalência. Os alunos também foram notificados que poderiam assistir quantas vezes acharem necessário os vídeos, na velocidade que eles preferiam.

Estágio 2: O segundo estágio compreende o momento em que os alunos deveriam estudar o conteúdo para se preparar para a aula. Considerando que os alunos foram instruídos no primeiro estágio em uma sexta-feira e a próxima aula da disciplina seria uma segunda-feira, os alunos teriam um final de semana (dois dias) para estudar o conteúdo.

Estágio 3: Na aula da segunda-feira, a professora solicitou aos alunos que eles fizessem perguntas sobre o conteúdo. Foi enfatizado que aquele momento era dedicado a resolução de dúvidas que eventualmente surgiram durante o estudo no estágio 2. Foram gastos aproximadamente 30 minutos, de uma aula que possui um total 110 minutos, para a resolução de dúvidas. Posteriormente, alguns exercícios sobre a aplicação do critério particionamento em classes de equivalência foram disponibilizados aos alunos. Esses exercícios deveriam ser realizados pelos estudantes durante a aula e o professor ficou disponível para resolver as dúvidas sobre a resolução do exercício. Ao final da atividade, os estudantes foram liberados.

Estágio 4: Na aula de sexta-feira, os alunos foram notificados que eles realizariam uma atividade individual, sem apoio do professor e de colegas e que não poderiam consultar o material da disciplina. Na ocasião, os alunos foram desafiados a realizar a atividade sobre a identificação de elementos requeridos. Após a resolução da atividade, os alunos foram notificados que a iniciativa fazia parte de um estudo experimental. Assim, eles foram apresentado ao objetivo do experimento e puderam informar se aceitariam disponibilizar os dados coletados para a análise do experimento.

Turma B – Grupo controle: O grupo controle participou de atividades em dois estágios distintos:

Estágio 1: Em uma terça-feira, no início da aula da disciplina, a professora explicou que antes de apresentar o critério de particionamento em classes de equivalência, ela gostaria de coletar algumas informações sobre os estudantes. Assim, o formulário de caracterização dos sujeitos foi aplicado. Após a aplicação, ela ministrou uma aula tradicional, em que a maior parte da aula foi ocupada por preleções apoiada por slides e exemplos de aplicações. Ao final das preleções, a professora apresentou algumas atividades que deveriam ser realizadas pelos alunos, mas que poderiam ser feitas em casa, depois da aula.

Estágio 2: Passando-se exatamente três dias, no início da aula, os estudantes foram notificados que deveriam realizar uma atividade individual e sem acesso a qualquer tipo de material de consulta. A atividade de identificação dos elementos requeridos foi aplicada para os estudantes desse grupo. Ao final da atividade, os alunos foram notificados que um estudo experimental estava sendo executado no contexto da disciplina e que cada aluno poderia decidir se disponibilizaria ou não os seus dados.

3.4.4 Análise dos Resultados

Após a execução do experimento, os dados do formulário de caracterização dos participantes e os dados associados à atividade foram analisados. Para apoiar a análise dos resultados, foram utilizadas análises descritivas e testes de estatística inferencial. Esta seção tem o propósito de apresentar essas análises.

Caracterização dos Participantes

A partir da execução do experimento, foi planejado que os 93 estudantes declarassem o interesse em disponibilizar ou não os dados para a análise. Muitos optaram por não disponibilizar, pois não se sentiram confortáveis em serem avaliados. Isso ocasionou uma redução no número de estudantes por grupo. Para contribuir com a redução do número de estudantes, nem todos eles participaram de todas as atividades do experimento, alguns faltaram aulas em que o experimento foi executado. Diante disso, a análise final contou com a participação de 79 estudantes, os quais participaram de todas as atividades e aceitaram disponibilizar os seus dados.

Dos 79 sujeitos, 35 (44%) estudaram o critério por meio de aulas baseadas no modelo tradicional de ensino, sendo caracterizados como integrantes do grupo controle. O restante (equivalente a 56%) estudou com uma iniciativa que implementou o modelo *flipped classroom*. Os sujeitos que participaram do estudo com *flipped classroom* foram

descritos como integrantes do grupo experimental. Visando compreender se os estudantes já haviam estudado ou possuíam algum tipo de experiência com o critério particionamento em classes de equivalência, uma questão do formulário de caracterização foi definida para esse fim. Como resultado, foi possível constatar que nenhum individuo possuía conhecimento e/ou experiência preliminar na área.

Efetividade

Conforme descrito anteriormente, a efetividade só pode ser observada a partir da análise da eficiência e da eficácia. A eficiência pode ser calculada por meio do tempo que o aluno "gasta" durante o reconhecimento dos elementos requeridos. Desse modo, para reconhecer a eficiência foi realizado um cálculo, tomando como base a seguinte fórmula:

$$Eficiencia_{(\mathrm{i})} = \frac{t_{\mathrm{MAX}} - t_{(\mathrm{i})}}{t_{\mathrm{MAX}}}$$

em que:

t = representa o tempo total gasto pelo aluno durante a realização da atividade;

i = representa o aluno, isto é, {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_n$};

t_{MAX} = represente o tempo máximo gasto pelo grupo de estudantes durante a realização da atividade.

A fórmula adotada para a eficiência foi a mesma que Souza (2017) utilizou em sua tese. Com as definições de t e t_{MAX}, caso o participante tenha despendido 100 minutos durante a identificação dos elementos requeridos, o tempo máximo foi atribuído à t. Assim, um participante que tenha despendido 50 minutos, recebe um valor de eficiência de 0,5, ou seja, 50% de eficiência naquela atividade. Por outro lado, o aluno que gastou o maior valor de tempo no grupo, recebe um valor de eficiência de 0,0, isto é, 0% de eficiência na atividade.

Em relação à eficácia, foram considerados o número de elementos requeridos definidos em um oráculo de teste, pelo pesquisador em conjunto com a orientadora e os elementos requeridos que foram identificados pelos participantes. Com base nisso, esses dados foram aplicados em uma fórmula que calcula a eficácia do participante em reconhecer os elementos requeridos.

$$Eficacia_{(\mathrm{i})} = \frac{n_{(\mathrm{i})}}{n_{\mathrm{TOTAL}}}$$

em que:

n = representa o número de elementos requeridos que foram identificados corretamente, isto é, que estavam previstos no oráculo e foram reconhecidos;

i = representa o aluno, isto é, {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_n$};

n_{TOTAL} = representa o número de elementos requeridos previstos pelo oráculo de teste.

Essa fórmula também foi baseada no cálculo para identificar a eficácia do estudante, utilizada no estudo de Souza (2017). Assim, o estudante que conseguiu identificar todos os elementos requeridos pelo critério que estavam previstos pelo oráculo, recebe um valor de eficácia "1,0"ou seja, 100% de eficácia. Em contraponto, se o aluno não conseguiu identificar corretamente qualquer um dos elementos requeridos, ele recebe um valor de eficácia de 0,0, ou seja, 0% de eficácia.

A partir disso, baseando-se no estudo de Souza (2017), tem-se a fórmula para calcular a efetividade, a saber:

$$Efetividade_{(i)} = Eficiencia_{(i)} * Eficacia_{(i)}$$

em que:

i = representa o aluno, isto é, {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_n$};

Eficiência = representa a eficiência do aluno em identificar os elementos requeridos para satisfazer o critério;

Eficácia = representa a eficácia do aluno em identificar os elementos requeridos para satisfazer o critério.

Estatística Descritiva

Na Tabela 16 são confrontados os principais resultados obtidos com respeito à eficiência, eficácia e efetividade. Eles foram contrapostos com base em medidas de posição, mais especificamente o valor mínimo, valor máximo, média, intervalo entre quartis. Também foi calculado o desvio padrão amostral para cada grupo, como medida de dispersão.

Observando as médias expostas na Tabela 16 relacionadas à eficiência, nota-se que o grupo que estudou com *flipped classroom* obteve melhores resultados, mas não significativamente distintos ao do grupo controle. Os demais valores relacionados a medida de posição também revelam proximidade. Todavia, o desvio padrão amostral associado a essa variável revela que o grupo experimental apresentou uma variação maior ao grupo controle. Isso indica que alguns alunos foram mais eficientes ao identificar os elementos requeridos enquanto outros alunos foram pouco eficientes.

Os valores de eficácia do grupo experimental são superiores aos valores de eficácia do grupo controle. As medidas de posição revelam que, em média, a eficácia foi duas vezes maior no grupo experimental. É possível observar que 50% dos valores de eficácia do grupo controle foram inferiores a 7%, enquanto que 50% dos valores de eficácia do grupo experimental foram superiores a 54%. Observando os valores mínimos de eficácia, notou-se que alguns estudantes do grupo controle não conseguiram identificar corretamente os elementos requeridos para satisfazer o critério.

Após uma análise comparativa entre as duas variáveis que caracterizam a efetividade, em síntese, observa-se que o grupo experimental apresentou uma maior efetividade em realizar a atividade ao qual eles foram expostos. O estudante que apresentou o melhor resultado para efetividade foi integrante do grupo experimental. Os valores da mediana revelam uma ampla diferença entre a efetividade do grupo experimental com o grupo controle, uma vez que 50% dos estudantes que estudou com *flipped classroom* obtiveram efetividade superior a 28%, enquanto que pelo menos 50% dos estudantes do grupo controle obtiveram efetividade equivalente a 0%.

Em virtude de alguns estudantes terem apresentados 0% de eficiência ou 0% de eficácia, alguns estudantes do grupo controle e também do grupo experimental obtiveram 0% de efetividade. Por consequência disso, os valores dos grupos sofreram uma ampla variação. Apesar da ampla variação, o grupo controle apresentou uma variação maior nos resultados de efetividade em relação ao grupo experimental.

Tabela 16 – Estatística descritiva dos valores de eficiência, eficácia e efetividade dos participantes

Variáveis	Grupo	Média	Desvio-Padrão	Mínimo	Máximo	Mediana	Q_1	Q_3
Eficiência	Controle	49%	25%	0%	93%	55%	31%	67%
	Experimental	51%	17%	0%	83%	54%	39%	63%
Eficácia	Controle	26%	30%	0%	100%	7%	0%	50%
	Experimental	53%	22%	14%	100%	54%	36%	64%
Efetividade	Controle	11%	16%	0%	57%	0%	0%	17%
	Experimental	27%	14%	0%	63%	28%	14%	34%

Fonte: Elaborada pelo autor.

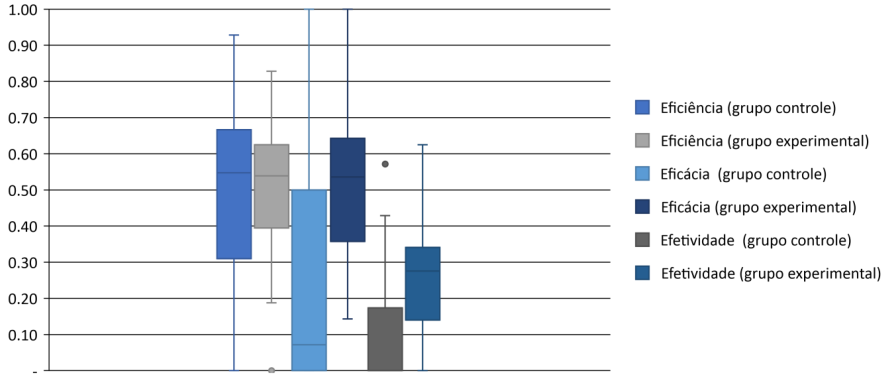
Para complementar a análise, foram gerados gráficos do tipo *box plot*. Na Figura 20 é possível observar, de forma resumida, os valores de eficiência, eficácia e efetividade

dos estudantes em reconhecer os elementos requeridos de um projeto de software para satisfazer o critério de teste. Em relação à eficiência, o gráfico revela uma maior amplitude nos dados do grupo controle que os dados do grupo de alunos que estudou com *flipped classroom*. Também foi possível observar que o grupo experimental conteve um valor de eficiência considerado *outlier* em relação aos seus integrantes.

Em relação à eficácia, observa-se que a medida de tendência central do grupo experimental supera o valor correspondente ao terceiro quartil do grupo controle. Analisando os *box-plots*, nota-se a existência de valores assimétricos nos dois grupos. Enquanto que o grupo controle apresentou uma assimetria positiva, o grupo experimental apresentou uma assimetria negativa. Assim, o *box plot* correspondente ao grupo controle indica que os valores que mais se repetem são menores que a mediana, que por sua vez é inferior à eficácia média. Em contrapartida, os valores que caracterizam as medidas de posição de eficácia do grupo experimental revelam que o valor da média é inferior a mediana, que por sua vez é inferior aos valores que mais se repetem no grupo (moda).

Finalmente, a efetividade como um todo foi analisada. Como é possível observar, os grupos apresentam níveis médios diferentes de efetividade, sendo que o grupo experimental apresenta uma efetividade superior ao grupo controle. Nessa perspectiva, a medida de tendência central do grupo controle é inferior a do grupo experimental. Além disso, também é possível observar que há uma presença de *outlier*, desta vez nos dados do grupo controle.

Figura 20 – *Box plot* para eficiência, eficácia e efetividade



Fonte: Elaborada pelo autor.

Testes de hipótese

Após a análise descritiva dos dados, visando averiguar se os alunos que estudam com *flipped classroom* tem mais efetividade ao reconhecer os elementos requeridos de um software em teste que os alunos que estudam com o modelo tradicional de ensino, foram

planejados e conduzidos testes de hipótese. Para tanto, inicialmente foi estabelecido um intervalo de confiança para α= 0,05.

Para apoiar a análise, foi utilizado o software BioEstat (versão 5.3). Nesse sentido, a normalidade dos dados foi verificada por meio do Teste de Shapiro-Wilk. A partir desse teste, em caso de identificação de distribuição dos dados de forma gaussiana, para dados não paramétricos, foi definido o uso do Teste t de Student. Em caso de distribuição anormal dos dados, foi definido o uso do Teste de Mann-Whitney.

Inicialmente, foram analisados os valores associados à eficiência. Constatou-se que tanto os valores de eficiência do grupo experimental quanto os valores do grupo controle possuem uma distribuição normal, uma vez que p-value para eficiência do grupo experimental foi igual a 0,2537 (isto é, p-value $> \alpha$) e o valor de p para eficiência do grupo controle foi de 0,3186 (isto é, p-value $> \alpha$). A partir disso, o Teste t de Student foi aplicado, revelando que não existe diferença significativa entre a eficiência do grupo controle com a eficiência do grupo experimental ao identificar os elementos requeridos. A Tabela 17 apresenta os dados completos da análise.

Tabela 17 – Teste t de Student - eficiência

	Grupo controle	**Grupo experimental**
Tamanho da amostra	35	44
Média	0,4883	0,5141
Variância	0,0604	0,0283
	Heterocedasticidade	
Variância	0,0024	
t	-0,5303	
Graus de liberdade	57,71	
p-valor (unilateral)	0,2990	
p-valor (bilateral)	0,5979	

Fonte: Elaborada pelo autor.

Em relação aos dados que remetem a eficácia, a partir do Teste de Shapiro-Wilk, notou-que que os dados não seguem uma distribuição normal. O valor de p para o grupo controle foi equivalente a 0,0087 (isto é, p-value $< \alpha$), enquanto que p-value para o grupo experimental foi de 0,0100 (isto é, p-value $< \alpha$). Desse modo, o Teste de Mann-Whitney foi realizado (Tabela 18). A partir do teste, foi possível notar que existe uma diferença significativa na eficácia entre os grupos, uma vez que o valor de p foi igual a 0,0001.

Finalmente, para compreender melhor a efetividade, o teste de hipótese foi aplicado. O Teste de Shapiro-Wilk revelou que os dados de efetividade que correspondem ao grupo controle não seguem uma distribuição gaussiana, uma vez que p-value igual a 0,0081 (isto é, p-value $< \alpha$). Em contraponto, os dados de efetividade que compreendem o grupo experimental possuem uma distribuição gaussiana, dado que o valor de p é igual a 0,0693 (isto é, p-value $> \alpha$). Assim, foi utilizado o Teste de Mann-Whitney para realizar o teste

Tabela 18 – Teste de Mann-Whitney - eficácia

	Grupo controle	Grupo experimental
Tamanho da amostra	35	44
Soma dos Postos (Ri)	1031,5	2128,5
Mediana	0,07	0,54
U	401,50	
Z (U)	3,6368	
p-valor (unilateral)	0,0001	
p-valor (bilateral)	0,0003	

Fonte: Elaborada pelo autor.

inferencial. Com base nos resultados, expostos na Tabela 19, a hipótese nula foi rejeitada, indicando que existe diferença em termos de efetividade em relação à identificação dos elementos requeridos feita por alunos que estudam com *flipped classroom* e por alunos que estudam com o modelo tradicional.

Tabela 19 – Teste de Mann-Whitney - efetividade

	Grupo controle	Grupo experimental
Tamanho da amostra	35	44
Soma dos Postos (Ri)	964,5	2195,5
Mediana	0,00	0,28
U	334,50	
Z (U)	4,2981	
p-valor (unilateral)	0,000001	
p-valor (bilateral)	0,000001	

Fonte: Elaborada pelo autor.

Como foi identificada uma diferença significante do ponto de vista da estatística na variável de efetividade, foi necessário avaliar o significado dos resultados sobre as diferenças encontradas entres os valores de efetividade nos grupos. Assim, para calcular o tamanho do efeito, foi utilizado o Teste de Cohen *(d)*. Como resultado, obteve-se um valor de $(d) = 1{,}02$. Considerando a argumentação de Cohen (1992), isso significa que o tamanho do efeito é grande, ou seja, a diferença estatística é efetivamente significativa. Para a condução do cálculo, foi utilizada a seguinte fórmula:

$$d = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2}}}$$

em que:

$\overline{x_1}$ e $\overline{x_2}$ representam as médias do grupo experimental e do grupo controle;

s_1^2 e s_2^2 representam as variâncias amostrais do grupo experimental e do grupo controle;

n_1 e n_2 representam os tamanhos amostrais do grupo experimental e do grupo controle.

3.4.5 Síntese das Análises

Tendo em vista que o estudo por meio do modelo *flipped classroom* pode influenciar no desenvolvimento de certas habilidades nos alunos, a tarefa de reconhecer os elementos requeridos que satisfazem um determinado critério de teste foi ponderada e colocada sob investigação durante a condução do estudo experimental. Assim, criou-se o experimento partindo do pressuposto que os alunos que estudam com *flipped classroom* consegue identificar os elementos requeridos com mais efetividade que os alunos que estudam com outro modelo pedagógico. O modelo pedagógico usado para comparar os resultados foi o modelo tradicional, uma vez que ele é ainda muito comum no ensino de teste de software (SOSKA; MOTTOK; WOLFF, 2016).

A efetividade, por sua vez, é uma variável que é interligada com as variáveis de eficiência, eficácia. Assim, para compreender melhor o que havia sido estabelecido em uma questão de pesquisa durante a definição, essas outras variáveis foram consideradas. Em relação à eficiência, a estatística descritiva indicou índices maiores para o grupo de alunos que estudou com *flipped classroom*. Apesar disso, os valores do grupo controle estavam próximos aos do grupo experimental. Com o Teste t de Student, foi possível inferir que a eficiência em identificar elementos requeridos foi equivalente entre os grupos.

Por outro lado, a análise descritiva sobre a eficácia indicou que os alunos que estudam com *flipped classroom* realizam com mais eficácia o reconhecimento dos elementos requeridos que os alunos que estudam com o modelo tradicional. Esse valor reflete em quão bem os estudantes realizaram a atividade. O teste de hipótese comprovou que há uma diferença significativa dos valores de eficácia entre os grupos.

Finalmente, visando compreender se os alunos reconhecem os elementos requeridos da maneira que deve ser identificados, atingindo os objetivos traçados e utilizando o tempo disponível da melhor forma possível, a efetividade foi observada. Análise descritiva indicou que o grupo experimental obteve índices de efetividade superiores que o grupo controle, indicando que são mais eficientes e eficazes ao mesmo tempo. O teste de hipótese em conjunto com o Teste de Cohen *(d)*, revelou que realmente há um diferença significativa em termos de efetividade entre os grupos.

Durante a análise, com a finalidade de concluir a análise a respeito da variável efetividade, foram feitos teste de hipóteses. Para tanto, foram estabelecidas sub-questões de pesquisa para as variáveis de eficiência e eficácia. A Tabela 20 sumariza os achados.

Tabela 20 – Resumo das respostas às questões relacionada a efetividade

Questão de pesquisa	**Resposta**
QP1 - Os alunos que estudam com Flipped Classroom têm mais efetividade ao reconhecer os elementos requeridos de um software em teste que os alunos que estudam com o modelo tradicional de ensino?	Sim. O aluno que estuda com *Flipped Classroom* reconhece os elementos requeridos que satisfazem o critério de particionamento em classes de equivalência de um software sob teste com mais efetividade que o aluno que estuda com o modelo tradicional.
QP1.1 Os alunos que estudam com *Flipped Classroom* têm mais eficiência ao reconhecer os elementos requeridos de um software em teste que os alunos que estudam com o modelo tradicional de ensino?	Não. Os dados indicam que os alunos que estudam com *Flipped Classroom* apresentam o mesmo índice de eficiência que os alunos que estudam com o modelo tradicional.
QP1.2 Os alunos que estudam com *Flipped Classroom* têm mais eficácia ao reconhecer os elementos requeridos de um software em teste que os alunos que estudam com o modelo tradicional de ensino?	Sim. Foram encontradas evidências de que o aluno que estudam com *Flipped Classroom* têm mais eficácia ao reconhecer os elementos requeridos de um software.

Fonte: Elaborada pelo autor.

3.4.6 Ameaças à Validade

As ameaças à validade serão discutidas no contexto das quatro categorias elencadas por Cook e Campbell (1979):

Validade de conclusão: As conclusões foram obtidas por meio dos resultados, os quais foram analisados através de testes estatísticos. Esses testes se definidos e utilizados de maneira incorreta podem produzir conclusões errôneas. Assim, para mitigar uma possível ameaças aos resultados, além de análise descritiva nos dados, foram realizados teste de hipóteses. Adicionalmente, para mitigar possíveis ameaças na definição do adequado teste, foram analisadas as normalidades dos dados. Contou-se também com um especialista da área durante a interpretação dos dados.

Validade interna: As conclusões sobre as causas e seus efeitos no estudo podem ser influenciados por fatores que não são previstos no estudo. Assim, os efeitos provocados nas variáveis analisadas (eficiência, eficácia, efetividade) podem não ser resultantes dos fatores definidos (com diferentes tratamentos) e sim atribuída a outras causas, como o projeto de software selecionado para o experimento ou o critério definido. Visando mitigar as ameaças que permeiam o experimento, algumas medidas foram tomadas.

A especificação do software foi selecionada porque retrata uma função em particular que recebe dados de entrada que podem ser classificados em diferentes subdomínios.

Em razão disso, esse projeto vem sendo utilizado por professores em diferentes universidades em todo o mundo, fazendo parte de livros específicos da área (JORGENSEN, 2013). A função mencionada na especificação do projeto é utilizada também em estudos experimentais da área (KHALSA; LABICHE; NICOLETTA, 2016).

Em relação ao critério utilizado no experimento, ele foi definido por ser relativamente simples (PASCHOAL; SOUZA, 2018a), é um critério que é ensinado pelos professores que ministram disciplinas de engenharia de software e/ou teste de software em cursos de Computação no Brasil (PASCHOAL; SOUZA, 2018b), além de oferecer condições para que um subconjunto de dados seja definido (FABBRI *et al.*, 2016).

Validade de constructo: Os instrumentos utilizados no experimento podem oferecer ameaças à validade do estudo. Nessa perspectiva, os dados relacionados ao tempo gasto pelo aluno para identificar os elementos requeridos, bem como, os elementos requeridos que foram identificados pelo aluno refletem em uma ameaça associada ao viés de um único método. Foi definida apenas uma atividade para os estudantes realizarem (identificar os elementos requeridos e anotar o tempo). É possível que a utilização de mais de uma atividade, como foi o caso do estudo realizado por Souza (2017), possa oferecer resultados mais precisos. Em razão do experimento ser executado em uma disciplina com um cronograma definido e prazos limitados, não foi possível realizar mais de uma atividade.

No estudo também foi utilizado apenas um único critério. É possível que o uso de diferentes critérios ofereça outros resultados para as variáveis analisadas. Apesar disso, o critério é direcionado ao projeto de software que foi definido e utilizado durante a execução.

Validade externa: Este experimento foi planejado e executado visando possibilitar e estimular sua replicação para comparar diferentes critérios e técnicas de teste. Todavia, alguns elementos podem oferecer ameaças à generalização dos dados obtidos para outro contexto. O projeto de software utilizado e o critério definido, por exemplo, podem não ser representativos de todos os tipos de atividades executados em cenários reais de ensino. Apesar disso, eles foram definidos de uma maneira cautelosa, visando possibilitar uma generalização dos resultados. Desse modo, a generalização dos resultados para estudantes em situações de aprendizagem semelhantes a utilizada no contexto do experimento não é uma ameaça para este estudo.

3.4.7 Apresentação e Empacotamento

Visando divulgar os resultados do estudo e ampliar o acesso ao conteúdo, um artigo científico que relata o planejamento, o processo de execução e os resultados obtidos será

escrito e submetido a um veículo de divulgação científica indexado. Busca-se disponibilizar o estudo para acesso em uma das principais bases de dados que indexam artigos da área de Computação, como a ACM, IEEE, e Scopus.

Todo o material utilizado, dados coletados e análises realizadas foram disponibilizados em um repositório de dados em formato de pacote de laboratório. Esse repositório é uma infraestrutura baseada em *web* de acesso público. Busca-se com o empacotamento facilitar futuras replicações e novas análises. O pacote de laboratório para este experimento pode ser obtido por meio do acesso ao seguinte endereço: <http://bit.ly/2FSw5qj>.

3.5 Considerações Finais

Este capítulo apresentou três estudos experimentais conduzidos a fim de avaliar *flipped classroom* enquanto um modelo pedagógico para o ensino de teste de software. Para tanto, foram analisadas variáveis que permeiam os processos de ensino-aprendizagem, a saber: motivação, desempenho acadêmico, aprendizado, carga de trabalho, preferência do aluno e efetividade. Visando aproximar o estudo do contexto real de ensino, os experimentos foram executados no contexto de disciplinas de engenharia de software.

Durante as análises do experimento, foram observadas as medidas de posição, medidas de dispersão e utilizados testes de hipótese, bem como, variáveis qualitativas a fim de obter a opinião dos estudantes sobre o modelo pedagógico *flipped classroom*. Em síntese, os resultados são promissores ao modelo, mostrando as possibilidades, ganhos e limitações.

Contudo, durante a realização dos experimentos, algumas limitações surgiram, em especial relacionadas a desistência dos participantes. Em razão dos experimentos terem sido executados em mais de uma aula, alguns estudantes não participaram de todas as atividades. Acredita-se que o ideal é que o experimento fosse executado em um mesmo dia, evitando que alunos participem parcialmente.

Outro ponto importante a ser considerado é que o impacto de utilizar *flipped classroom* no contexto de uma disciplina de engenharia de software não foi analisado. É possível que alguns resultados obtidos possam estar relacionados ao desinteresse dos alunos nessa disciplina. Espera-se poder confrontar esses resultados com outros, advindos de outras disciplinas de Computação.

Com base em toda a experiência adquirida por meio da definição de estratégias para a implantação do modelo durante os experimentos, uma abordagem que visa subsidiar o planejamento e a implementação desse modelo durante o ensino de teste de software foi proposta. O próximo capítulo apresenta a proposta de uma abordagem para flexibilizar e oportunizar o uso de *flipped classroom* no ensino de teste de software.

CAPÍTULO

4

FLIPCLASS-STT

4.1 Considerações Iniciais

A implementação efetiva e eficiente do modelo *flipped classroom*, apesar de ser relevante ao ensino de determinados conteúdos e áreas, pode ser desafiadora para o professor, especialmente quando esse está acostumado a ensinar o conteúdo utilizando estratégias pedagógicas fundamentadas no modelo tradicional de ensino. De fato, abandonar as preleções feitas em sala de aula para se tornar um orientador das individualidades do aluno, durante a aula, exige do professor dedicação e tempo que muitas vezes ele não tem a sua disposição. Estudos indicam que a carga de trabalho do professor aumenta ao adotar o modelo *flipped classroom*, pelo menos na primeira vez em que ele for implantado (BERGMANN; SAMS, 2012).

Para utilizar o modelo *flipped classroom*, o professor terá que preparar aulas predominadas por atividades que buscam favorecer o desenvolvimento de habilidades nos alunos. Essas atividades podem estimular a prática sobre o conteúdo e, inclusive, proporcionar momentos que permitem ao aluno que ele seja autor do conteúdo. Os materiais didáticos, ao mesmo tempo, precisam ter uma natureza que oportuniza a aprendizagem dos alunos, tendo como característica uma boa qualidade. Por consequência disso, os professores podem encontrar dificuldades na confecção adequada dos materiais. O conteúdo deve enriquecer os momentos práticos, assim, eles precisam ser definidos cautelosamente. É possível que determinados conteúdos não sejam adequados de serem ensinados com o modelo.

Em geral, assume-se que o modelo *flipped classroom* prevê três momentos de aprendizagem, já mencionados no Capítulo 1. O primeiro momento, conhecido por "antes da aula", contempla as atividades passivas que devem ser realizadas pelos alunos antes dos mesmos terem contato com o professor em sala de aula. É nesse momento que os alunos

irão ter o acesso preliminar ao conteúdo. Assim, materiais didáticos e atividades devem ser oferecidas de modo que o aluno, sem conhecimento prévio, consiga estudar sem ter problemas. Nesse sentido, um material didático adequado deve ser oferecido ao aluno, de modo a facilitar a aprendizagem. O conteúdo abordado também pode criar casualidades não previstas. É possível, por exemplo, que os alunos não consigam entender o conteúdo, por ele ser muito abstrato ou por falta de qualidade.

O segundo momento, intitulado "durante a aula", prevê o encontro da turma de alunos com o professor. Nessa ocasião, os estudantes devem expor as suas dúvidas, dificuldades ao estudar o conteúdo (ou um tópico do conteúdo), para que o professor possa aproveitar o momento para saná-las, ajudando o aluno a compreender melhor o assunto. Também são previstas atividades práticas, as quais devem ser acompanhadas pelo professor, que neste momento assume um papel de supervisor, devendo orientar os alunos e solucionar as dúvidas. Atividades em grupos devem ser estimuladas, assim, os alunos podem construir o conhecimento coletivamente.

Nessa perspectiva, o professor deve definir o tempo necessário para solucionar as dúvidas dos alunos, o qual pode ser facilmente ampliado ou sintetizado, em decorrência do número de dificuldades relatadas pelos participantes. A atividade prática precisa desafiar os alunos, porém, deve ser condizente com que o aluno estudou. A resolução de problemas pode ser estimulada nesse momento, uma vez que possui como característica a capacidade de flexibilizar o aprendizado, em que os alunos aos poucos conseguem solucionar os problemas e construir novos conhecimentos. Atividades de laboratório também são interessantes nesse momento, uma vez que oferecem subsídios ao desenvolvimento de habilidades práticas na área.

O terceiro momento, não menos importante, conhecido como "depois da aula", pode ser dedicado a revisão do conteúdo, visando estimular o aluno a melhorar o conhecimento adquirido. Pode ser utilizado para oferecer oportunidade de treinamento, em que os alunos são incentivados a participar de testes de conhecimento. Para esse momento, jogos sérios podem ser utilizados, especialmente aqueles com desafios divididos em níveis. Assim, depois do estudo, o aluno pode jogar uma ou duas fases de um jogo. Por outro lado, o momento pode ser utilizado para que o aluno consiga estudar um conteúdo preliminar para a próxima aula.

Todas as decisões tomadas, influenciarão na implementação do modelo. Em razão disso, nesta dissertação, com o propósito de contribuir com a implementação ou pelo menos com a tomada de decisão por parte do professor ao definir a adoção o modelo, foi estabelecida uma abordagem para apoiar a implementação do modelo *flipped classroom* em aulas de teste de software. A abordagem pode ser utilizada por professores que ministram teste em disciplinas específicas ou disciplinas não específicas, como engenharia de software. Nessa perspectiva, no decorrer deste capítulo, todas as atividades realizadas

para a definição da abordagem serão descritas. Com base nessas atividades, são oferecidas algumas direções que contribuem com o estabelecimento de uma argumentação para a segunda questão de pesquisa abordada durante a dissertação, que é "Como implementar o modelo *flipped classroom* no ensino de teste de software?".

4.2 Estabelecendo a abordagem FlipClass-STT

A experiência adquirida por meio do planejamento e condução de estudos experimentais, contribuiu para a identificação dos elementos e das estratégias que precisam ser adotadas em uma aula ou em um curso que é baseado no modelo *flipped classroom*. O estudo sobre os direcionamentos para planejamento e implantação do modelo descritos por Bergmann e Sams (2012) também ofereceu alguns subsídios. Nessa perspectiva, durante esta seção serão apresentadas as atividades realizadas para definir uma abordagem de apoio ao ensino de teste de software com *flipped classroom*, intitulada FlipClass-STT.

Salienta-se que para apoiar o estabelecimento da abordagem, inicialmente foi necessário definir quais os conteúdos de teste de software que seriam contemplados pela FlipClass-STT. A partir da definição dos conteúdos, os objetivos de aprendizagem para cada conteúdo foram determinados. Na sequência, ocorreu a seleção e o estabelecimento de recursos educacionais com a finalidade de satisfazer o conteúdo e os objetivos de aprendizagem. Dando prosseguimento, atividades foram identificadas.

4.2.1 Definição de conteúdo

Existem vários esforços por parte da comunidade de teste de software em estimular o professor a explorar diferentes mecanismos de apoio ao ensino e modelos pedagógicos nessa área. Apesar disso, não há um panorama que indique quais são os assuntos que estão sendo ensinados pelos professores que ministram disciplinas que compreendem a atividade de teste de software. A falta do reconhecimento dos conteúdos pode impactar no desenvolvimento equivocado de mecanismos de apoio e abordagens. Por consequência, mecanismos e abordagens correm o risco de ficarem subutilizadas.

Nesse sentido, visando mitigar a possível subutilização da abordagem, a primeira atividade realizada foi definir quais os conteúdos que seriam abordados na FlipClass-STT, tendo como base os conteúdos que são ensinados por professores da área. Para tornar explícito o que a comunidade de computação ensina sobre teste de software, um *survey* descritivo foi planejado e executado com professores atuantes em cursos de educação superior em computação no Brasil. O *survey* foi pré-definido seguindo o processo proposto por Kasunic (2005) para planejamento efetivo de *surveys* para a área de engenharia de software. Adicionalmente, os direcionamentos descritos por Kitchenham e Pfleeger (2008) foram considerados.

O processo proposto por Kasunic (2005) compreende sete etapas, mencionadas na Figura 21, a saber: (1) Identificação dos objetivos de investigação; (2) Identificação e caracterização do público – alvo; (3) Planejamento de amostragem; (4) Planejamento e escrita do questionário; (5) Execução piloto; (6) Distribuição do questionário (*survey*); e, por fim, (7) Análise dos resultados e escrita de um relatório. Cada uma das etapas será brevemente apresentada a seguir:

(1) Identificação dos objetivos de investigação: Na primeira etapa, os objetivos do *survey* são estabelecidos, buscando guiar o escopo do *survey*.

(2) Identificação e caracterização do público – alvo: Considerando os objetivos que foram definidos, é necessário determinar quem são os indivíduos que podem participar do estudo, isto é, os sujeitos que podem oferecer as respostas que auxiliarão a atingir os objetivos.

(3) Planejamento de amostragem: Nessa etapa é definida a estratégia para selecionar do público-alvo uma amostragem. Também são discutidas estratégias para estabelecer contato com os participantes.

(4) Planejamento e escrita do questionário: Os objetivos do *survey* devem ser transcritos em formato de questões. Elas irão compô-lo e devem ser escritas de modo criterioso, facilitando a interpretação do sujeito que faz parte da amostragem.

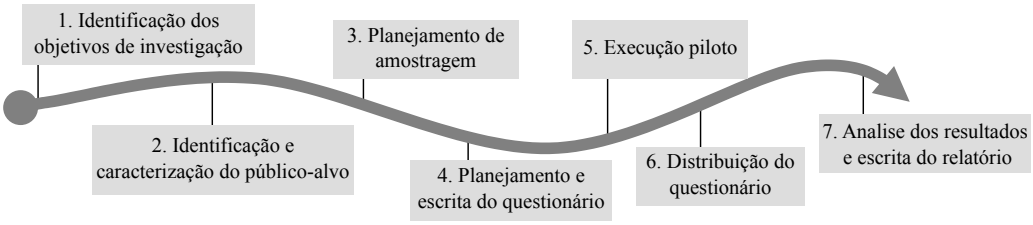
(5) Execução piloto: Nessa etapa, alguns integrantes que fazem parte do público-alvo são convidados a participar do *survey*. Eles precisam identificar eventuais problemas existentes no instrumento. Com base nisso, busca-se reconhecer melhorias que precisam ser feitas no *survey*, antes dele ser distribuído à amostragem.

(6) Distribuição do questionário (*survey*): Após os ajustes no *survey*, ele é encaminhado para os participantes que compõem a amostragem. Nesse sentido, busca-se coletar os dados para responder as questões que constituem o instrumento.

(7) Análise dos resultados e escrita de um relatório: Por fim, a última etapa que compreende o processo refere-se a análise dos resultados. Durante essa etapa, também é recomendado a elaboração de relatórios ou artigos que solidificam as descobertas.

No decorrer das próximas subseções serão apresentados detalhes do processo de planejamento e condução do *survey*.

Figura 21 – Processo adotado na condução do *survey*



Fonte: Adaptada de Kasunic (2005).

Identificação dos objetivos de investigação

O principal objetivo da condução do *survey* é o reconhecimento sobre o conteúdo de teste de software que é ensinado em cursos de graduação em computação no Brasil. Nesse sentido, procurou-se identificar as técnicas e os critérios ensinados, bem como os níveis em que a atividade é explanada pelos professores. A partir do esclarecimento das necessidades, o objetivo principal foi especificado por meio da abordagem GQM (Goal/Question/Metric) proposta por Basili (1992), descrito na Tabela 21. Para apoiar a definição das questões e métricas foram utilizados alguns artefatos de apoio como: o guia SWEBOK (BERTOLINO; MARCHETTI, 2014) e uma ontologia de teste de software denominada OntoTest (BARBOSA; NAKAGAWA; MALDONADO, 2006).

Tabela 21 – Objetivos do *survey*

Objetivo (*Goal*)	**Questão (*Question*)**	**Métrica (*Metric*)**
Reconhecer os conteúdos que são ensinados.	1. Quais os níveis de teste que são ensinados?	1. Conjunto de opções com os principais níveis de teste.
	2. Quais técnicas de teste são ensinadas?	2. Conjunto de opções com as principais técnicas de teste de software.
	3. Quais os critérios de cada técnica de teste de software são ensinados?	3. Conjunto de opções de critérios agrupados conforme as principais técnicas de teste.
	4. Selecione e/ou indique outros temas relacionados que são ensinados.	3. Conjunto de opções com outros temas.

Fonte: Elaborada pelo autor.

Identificação e caracterização do público-alvo

O público-alvo definido para participar do *survey* é composto por professores que possuem experiência com o ensino de teste de software, adquirida por meio de aulas ministradas em disciplinas de teste de software ou engenharia de software em cursos superiores de computação. Esses professores devem ter atuado em cursos autorizados e reconhecidos pelo Ministério da Educação. Após a definição do público-alvo, foi planejado o processo de obtenção de amostras.

Planejamento de amostragem

O processo de obtenção de amostras da população, foi planejado para seguir uma metodologia que oferecesse subsídios para alcance de resultados da amostra que fossem representativos de toda a população. Assim, em um primeiro momento pretendeu-se utilizar uma amostragem probabilística, em que cada elemento da população tivesse a mesma probabilidade de ser selecionado para constituir a amostragem (BUSSAB; MORETTIN, 2013).

Como estratégia para identificar os professores atuantes foram localizados os cursos de graduação, por meio do sistema e-MEC[1] e entrar em contato com os coordenadores desses cursos para que estes passassem o contato dos professores que ministram o conteúdo de teste de software. Como engenharia de software é uma disciplina recomendada pelos currículos de referências, acredita-se que cada instituição de ensino deve oferecer a disciplina e, portanto, cada instituição deve possuir pelo menos um professor que já tenha ministrado o conteúdo de teste de software, mesmo que em uma disciplina não específica.

Foram identificados os cursos de graduação em computação (bacharelado) em atividades no Brasil. Em outubro de 2017, foi realizada uma busca no portal e-MEC a fim de localizar os cursos e os coordenadores vinculados a esses cursos, que resultou em 1438 cursos em atividade. A partir dos dados coletados sobre os cursos, foi possível observar que o coordenador do curso não estava sendo indicado em todos os cursos em análise, comprometendo a identificação. Além disso, os cursos que possuíam um coordenador cadastrado não possuíam o contato do coordenador, dificultando o processo de identificação para o possível contato. Essas adversidades acabaram dificultando a obtenção de amostras probabilísticas, como seria desejável. Desse modo, foi necessário aplicar processos não probabilísticos de amostragem.

Com a inacessibilidade a toda a população, recorreu-se às listas eletrônicas de comissões especiais[2], mantidas por profissionais e instituições participantes da Sociedade Brasileira de Computação. Essas listas possuem como membros pesquisadores e professo-

[1] Mais informações disponíveis em: <https://emec.mec.gov.br/>
[2] Mais informações podem ser obtidas em: <http://www.sbc.org.br/22-destaques/34-listas-eletronicas>

res da área de computação. Nesse sentido, o tipo de amostragem foi definido como não probabilística, selecionada por conveniência.

Planejamento e escrita do questionário

Nesta etapa do processo foram revistos os objetivos pré-definidos durante a primeira etapa do processo, bem como a forma como os dados dos participantes que compõem a amostra seriam coletados. Considerando que os participantes são assinantes das listas de *e-mails* das comissões mencionadas, foi definido que o *survey* seria disponibilizado eletronicamente, em formato de questionário eletrônico. Ele foi construído utilizando a ferramenta Google Forms[3]. O questionário *web* foi estruturado em seções.

A primeira seção contém uma apresentação do *survey* em que é descrito o seu objetivo, o público-alvo, esclarecimentos sobre a importância da participação de professores atuantes na área, as datas importantes que delimitam o prazo em que o *survey* estaria sendo disponibilizado para receber respostas. A seção também possui um link para um glossário[4] dos termos que aparecem no *survey*.

A segunda seção foi construída para coletar informações que caracterizam os participantes, são elas: nome do participante, afiliação, estado da instituição de ensino e a maior titulação acadêmica do professor. Esses dados também são utilizados para validação dos participantes, visto que, por decorrência do questionário ser aberto, corria-se o risco desses participantes não serem professores atuantes em cursos de computação. Assim, a partir dessas informações, foi possível validar a autenticidade recorrendo-se a plataforma Lattes[5]. Nessa plataforma, os currículos dos professores que responderam o *survey* foram identificados e analisados, visando confrontar os dados que os mesmos informaram nesta seção.

Na sequência, uma seção teve o intuito de coletar informações se o curso em que cada respondente atua possui uma disciplina específica destinada ao ensino de teste de software. Essa seção foi definida para possibilitar uma análise dos dados por grupos, em que um grupo é constituído por respostas de professores que ministram o conteúdo em disciplinas específicas e o outro grupo representa as respostas sobre o conteúdo ministrado em disciplinas não específicas.

Finalmente, outra seção aborda as questões que foram estipuladas para recolher informações como os níveis, as técnicas e os critérios de teste de software. Também foi escrita uma questão para os professores indicarem outros assuntos que são ensinados. Essas questões foram definidas com apoio de um especialista na área. Após a definição e construção do questionário eletrônico, foi realizada uma execução piloto, visando analisar

[3] Mais informações podem ser obtidas em: <https://www.google.com/forms/about/>
[4] O glossário encontra-se disponível em: <http://bit.ly/2NsPHlO>
[5] Mais informações podem ser obtidas em: <http://lattes.cnpq.br/>

a validade do instrumento.

Execução piloto

Kasunic (2005) menciona sobre a importância da execução piloto do *survey*, a fim de detectar os problemas existentes, verificar se as perguntas são compreensíveis, se as perguntas certas foram feitas para atingir o objetivo, e quanto tempo os participantes levam para completarem o questionário. Nesse sentido, o estudo piloto contribui com a qualidade do instrumento.

Uma forma de verificar a validade de um instrumento é por meio da avaliação do seu conteúdo (COOK; BECKMAN, 2006), que por sua vez consiste em uma medida subjetiva para identificar o quanto os itens de um instrumento são apropriados, por meio da opinião de especialistas (LITWIN, 1995). Hauck, Wangenheim e Wangenheim (2011) prepararam quatro questões abertas para avaliar o conteúdo de *surveys*, em execuções piloto. Essas questões foram utilizadas no estudo piloto realizado, são elas:

- (1) o questionário contém tudo que é esperado para contemplar o seu objetivo?
- (2) o questionário contém quaisquer informações não desejáveis ou desnecessárias ao contexto e objetivo da pesquisa?
- (3) você conseguiu compreender adequadamente as perguntas?
- (4) existe algum erro ou inconsistência no questionário?

Um grupo de professores e alunos de doutorado do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, atuantes na área de teste de software, foram convidados por *e-mail* para participar do teste piloto. Esses especialistas foram escolhidos pelos critérios de disponibilidade e proximidade com o grupo onde esta pesquisa foi desenvolvida.

Cinco especialistas (dois doutorandos e três doutores) participaram do estudo piloto, respondendo as questões definidas por Hauck, Wangenheim e Wangenheim (2011) e emitindo *feedback* sobre o *survey*. A avaliação dos especialistas foi positiva, com sugestões para: diminuir a quantidade de perguntas, deixando algumas não obrigatórias; reduzir o tempo estimado para responder o *survey* para 10 minutos (ao invés de 30 minutos) e sugestão para incluir caixas de seleção para algumas perguntas.

Distribuição do questionário

A partir do parecer dos especialistas, ajustes foram feitos no *survey* para que ele pudesse ser distribuído para a amostragem. O questionário foi distribuído por meio de

e-mails cadastrados nas listas das comissões mencionadas. Essa distribuição ocorreu entre 13 e 30 de novembro de 2017. Depois do fim do período em que o questionário ficou disponível para recebimento de respostas, os dados coletados foram analisados, visando estabelecer como os conteúdos de teste de software são ensinados.

Resultados

O *survey* foi respondido por 52 professores, atuantes em diferentes instituições de ensino, de diferentes estados brasileiros, que abrangem as cinco regiões do país. O maior número de participantes atua na região Sudeste (38%), seguido pela região Sul (36,54%), região Nordeste (15,38%), Centro-Oeste (5,77%) e por último a região Norte (3,85%). Apesar da pesquisa ter conseguido respondentes de diferentes regiões, não foi possível obter a participação de professores de todos os estados do Brasil. Assim, uma análise por estado não foi possível. Também não foi possível caracterizar como teste de software é ensinado por região do Brasil.

Para apoiar a caracterização, foi solicitado aos participantes que informassem a sua maior titulação acadêmica. A partir disso foi possível observar que a maioria dos participantes possui doutorado (61,54%). Do total de participantes, 36,54% possuem a titulação de mestre e 1,92% de especialista.

Levando em consideração os trabalhos conduzidos por Silva, Müller e Bernardi (2011), Benitti e Albano (2012) e Valle, Barbosa e Maldonado (2015), foi definida uma questão para averiguar se os participantes ministram o conteúdo de teste de software em uma disciplina específica para o conteúdo -– em que apenas teste de software é ensinado – ou uma disciplina não específica como engenharia de software. Como resultado, 28 participantes (54%) relataram que ministram o conteúdo em disciplina não específica, e 24 (46%) lecionam o conteúdo em uma disciplina específica de teste de software.

Quando questionados sobre quais são os níveis de teste ensinados, os resultados da pesquisa apontam que teste de unidade, teste de integração e teste de sistema são ensinados tanto pelos professores que ministram disciplinas específicas de teste quanto professores que ministram teste em uma disciplina não específica. A Tabela 22 apresenta os resultados completos, considerando o nível de teste, número de professores que ministra disciplina específica e número de professores que ministra disciplina não específica.

Todos os professores ensinam teste em nível de unidade, mais da metade ensina teste de integração e teste de unidade. Com os resultados, é possível observar que a maioria dos professores concentra os seus esforços em ensinar os alunos a realizar teste em pequenas unidades de software. Acredita-se que um fator que pode causar esses resultados está associado com a complexidade de se trabalhar com testes de software que consigam contemplar um sistema completo. Além disso, alguns professores relataram que outros níveis são ensinados, tais como teste de aceitação, teste de regressão, dentre outros.

Tabela 22 – Níveis de teste de software ensinados

Nível de teste	Disciplina específica	%	Disciplina não específica	%
Teste de unidade	24	100	28	100
Teste de integração	23	95,83	23	82,14
Teste de sistema	17	70,83	22	78,57
Teste de aceitação	3	12,5	1	3,57
Teste de carga	1	4,17	0	0
Teste de desempenho	0	0	1	3,57
Teste de escalabilidade	0	0	1	3,57
Teste de estresse	1	4,17	2	7,14
Teste de lançamento	0	0	1	3,57
Teste de regressão	1	4,17	0	0
Teste de usabilidade	1	4,17	0	0

Fonte: Elaborada pelo autor.

Em relação às técnicas de teste de software que são ensinadas, destacam-se o teste funcional e o teste estrutural. Ambas as técnicas são ensinadas por mais da metade dos professores que ministram o conteúdo em uma disciplina específica e disciplina não específica. No caso da técnica de teste funcional, ela é ensinada por todos os professores que ministram o conteúdo em uma disciplina específica, mas nem por todos os que ministram em disciplina não específica. Observa-se que quando há uma disciplina específica para o conteúdo, a técnica de teste baseado em defeito é ensinada por mais da metade dos professores. Em contrapartida, quando a disciplina não é específica, mais da metade dos professores deixa de ensiná-la. A Tabela 23 apresenta os resultados detalhados.

Tabela 23 – Técnicas de teste de software ensinadas

Técnicas de teste	Disciplina específica	%	Disciplina não específica	%
Teste funcional	24	100,00	25	89,29
Teste estrutural	19	79,17	20	71,43
Baseado em defeitos	18	75,00	8	28,57

Fonte: Elaborada pelo autor.

Apesar de o teste funcional ser ensinado por 100% dos professores que ministram o conteúdo em uma disciplina específica, nem todos ensinam os critérios dessa técnica. Dos 24 professores, apenas 20 mencionam os critérios dessa técnica. Os dados revelaram que o critério Particionamento em Classes de Equivalência é ensinado por 83,33% dos 24 professores, sendo o critério do teste funcional que se sobrepõem aos demais. Esse critério é seguido por Análise do Valor Limite que é lecionado por 79,17% dos 24 participantes. De maneira semelhante, a técnica de teste funcional e seus respectivos critérios aparecem

em maior intensidade que as outras técnicas quando o conteúdo é ensinado em disciplinas não específicas de teste.

Teste estrutural é a segunda técnica de teste com maior repercussão nos dados obtidos, conteúdo ensinado por mais da metade dos professores, tanto por aqueles que ministram o conteúdo em disciplina específica quanto por aqueles que não ministram. Os dois principais critérios dessa técnica (Critérios baseados em Fluxo de Controle e baseados em Fluxo de Dados) são ensinados por pelo menos metade dos professores que ministram o conteúdo em disciplina específica. Quando a disciplina não é específica, apenas os Critérios baseados em Fluxo de Controle são ensinados.

Em relação aos critérios do teste baseado em defeitos, quando teste é ensinado em disciplina específica, mais da metade dos participantes (75%) revelaram que ensinam o critério teste de mutação. Em contrapartida, quando essa técnica é ensinada em uma disciplina não específica, mais da metade (71,43%) dos participantes revelou que não a ensina.

A Tabela 24 apresenta os resultados em relação aos critérios de teste de software que são ensinados, por técnica de teste.

Tabela 24 – Critérios de teste ensinados, agrupados por técnica

Técnica de teste	Critério de teste	Disciplina específica	%	Disciplina não específica	%
Funcional	Particionamento em classes de equivalência	20	83,33	19	67,86
	Análise do valor limite	19	79,17	16	57,14
	Grafo causa-efeito	11	45,83	10	35,71
	Teste funcional sistemático	11	45,83	8	28,57
Estrutural	Critérios baseados em fluxo de controle	19	79,17	18	64,29
	Critérios baseados em fluxo de dados	12	50,00	11	39,29
Baseado em defeitos	Teste de Mutação	18	75,00	8	28,57

Fonte: Elaborada pelo autor.

Para finalizar a seção de reconhecimento sobre os assuntos de teste de software que são ensinados, os participantes tiveram que indicar outros temas relacionados que

eles ensinam. Observa-se na Tabela 25 os outros assuntos que são ensinados. Por meio da tabela, nota-se que os conteúdos depuração de software, outras atividades de VV&T, e terminologia de teste são abordados em uma maior porcentagem por professores que atuam em disciplinas não específicas de teste.

Tabela 25 – Critérios de teste ensinados, agrupados por técnica

Outros assuntos ensinados	**Disciplina específica**	**%**	**Disciplina não específica**	**%**
Depuração de software	6	25,00	9	32,14
Documentação de teste	12	50,00	12	42,86
Ferramentas de teste	19	79,17	22	78,57
Geração automática de dados de teste	7	29,17	4	14,29
Outras atividades de verificação, validação e teste (VV&T)	13	54,17	18	64,29
Processo de teste	11	45,83	12	42,86
Terminologia de teste	16	66,67	19	67,86

Fonte: Elaborada pelo autor.

Apresentação dos resultados

O planejamento do *survey* e os resultados obtidos com a sua execução foram reportados em formato de um artigo científico. Ele foi apresentado e publicado nos anais do 17º Simpósio Brasileiro de Qualidade de Software, com o título de "A Survey on Software Testing Education in Brazil". Neste artigo também são discutidas outras questões consideradas relevantes para a temática de ensino de teste de software. Salienta-se que o artigo recebeu destaque pela comissão organizadora do evento, considerando-o como o melhor artigo da trilha de educação em qualidade de software.

Contribuições do survey à definição do conteúdo

A partir dos resultados obtidos por meio *survey* é possível ter uma visão geral sobre quais os conteúdos abordados pelos professores, em cursos de computação. Nesse sentido, foi definido que a abordagem deve contemplar a técnica de teste funcional em conjunto com os critérios particionamento em classes de equivalência e análise do valor limite. Também deve abranger critérios da técnica de teste estrutural, em especial critérios baseados em fluxo de controle. Esses critérios e técnicas foram definidas por serem ensinadas por pelo menos 50% dos participantes do *survey*.

Buscando apoiar o ensino de critérios menos abordados pelos professores, em especial pelos professores que ensinam teste de software em conjunto com outros temas, em disciplinas não específicas, foi definido que critérios menos ensinados também seriam

contemplados pela abordagem. Assim, critérios baseados em fluxo de dados da técnica de teste estrutural e o critério teste de mutação também foram levados em consideração. Portanto, a abordagem contempla os principais critérios e técnicas de teste de software descrito por Delamaro, Jino e Maldonado (2016).

Com base no que foi descrito, a abordagem FlipClass-STT foi definida para contemplar aspectos práticos da atividade de teste de software, não contemplando aspectos mais básicos como terminologias de teste ou aspectos mais complexos como a geração automática de dados de teste. Em virtude de abranger aspectos práticos, outro tópico que é favorecido pela abordagem é o uso de ferramentas de teste.

4.2.2 Definição dos objetivos de aprendizagem

Flipped classroom pode ser considerado um modelo pedagógico que se baseia em diferentes teorias e tendências de aprendizagem (BERGMANN; SAMS, 2012) para definir uma estratégia de organização de ensino, em que atividades de aprendizagem (passivas e ativas) sofrem uma alternação de momentos. Assim, atividades passivas passam a ser realizadas fora da sala de aula e o momento em sala de aula é dedicado a condução de atividades ativas. Por ser uma estratégia de ensino, a partir da definição do conteúdo, é necessário definir com clareza o que é esperado que a instrução invertida promova nos alunos. Em outras palavras, define-se os objetivos de aprendizagem, cujo propósito é especificar quais as habilidades os estudantes devem adquirir ao final do processo.

Existe uma discussão na literatura sobre a necessidade de definir os objetivos de aprendizagem, uma vez que a prática pedagógica se orientada para atingir objetivos. Em geral, o professor deve estabelecer quais são os objetivos que o aluno deve atingir ao estudar. Somente após a definição, o professor conseguirá definir quais os materiais são adequados para que o aluno consiga as instruções em busca do objetivo. Ao fim do estabelecimento dos objetivos de aprendizagem, iniciativas para promover o desenvolvimento de competências podem ser planejadas e atividades que terão o intuito de averiguar se o aluno conseguiu atingir os objetivos pré-estabelecidos devem ser produzidas.

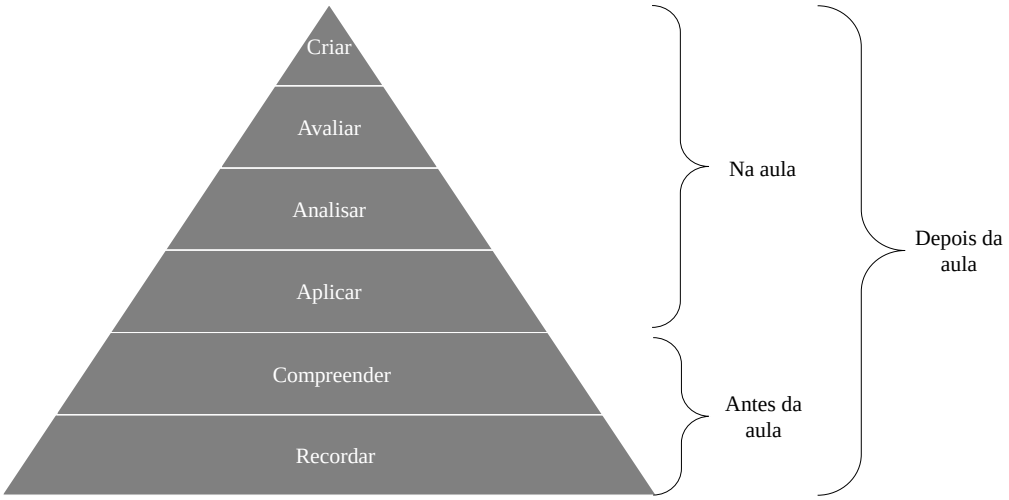
No estudo de Kiwelekar e Wankhede (2015), os autores fazem uma discussão sobre a importância da definição de objetivos de aprendizagem para disciplinas de Arquitetura de Software e ao discutir sobre os benefícios do estabelecimento de objetivos de aprendizagem, menciona que os processos educacionais (*i.e.*, instrução, aprendizagem e avaliação) podem ser alinhados de maneira eficaz. Assim, pode-se assimilar que os objetivos de aprendizagem são atrelados aos componentes do aprendizado (*e.g.*, conteúdo, materiais didáticos, metodologia, atividades e avaliação).

Os objetivos de aprendizagem vem sendo redigidos por meio de modelos. Acredita-se que o modelo mais conhecido é a Taxonomia de Bloom. Essa taxonomia refere-se à

divisão de objetivos educacionais em três partes: cognitiva, afetiva e psicomotora (ANDERSON; KRATHWOHL; AIRASIAN, 2001), sendo o domínio cognitivo bastante explorado durante a definição dos objetivos. O domínio cognitivo possui seis níveis: Avaliação, Síntese, Análise, Aplicação, Compreensão, Conhecimento. Em síntese, cada um desses níveis representa o que o sujeito deve aprender (ANDERSON; KRATHWOHL; AIRASIAN, 2001).

A Taxonomia de Bloom vem sendo utilizada para apoiar a definição de objetivos de aprendizagem em aplicação de *flipped classroom*. Fassbinder *et al.* (2014) mencionam que essa taxonomia, em especial a dimensão de domínio cognitivo, pode apoiar o planejamento didático e pedagógico. Teixeira (2013) complementa descrevendo que a taxonomia pode ser facilmente integrada ao modelo *flipped classroom*. A partir da integração, o autor cita a Figura 22.

Figura 22 – Taxonomia de Bloom e Flipped Classroom



Fonte: Adaptada de Teixeira (2013).

Teixeira (2013) salienta que a progressão dos seis domínios cognitivos ocorre nos três momentos previstos pelo modelo. Antes da aula, os alunos assistirão os vídeos instrucionais e/ou farão a leitura do material que foi indicado pelo professor. Nesse momento, eles recordarão e compreenderão os conceitos. Durante a aula, os estudantes devem aplicar, analisar, avaliar e criar artefatos a partir de atividades práticas e de apoio do professor. Posteriormente, depois da aula, os alunos continuarão revendo conceitos, aplicando-os, avaliando-os, buscando criar novos projetos e artefatos. Uma descrição sobre cada um dos níveis da taxonomia será apresentada a seguir:

Recordar: esse nível está relacionado a capacidade do aluno em reconhecer, reproduzir ou recordar uma determinada informação;

Compreender: esse nível está relacionado a capacidade do aluno em estabelecer um relação entre o conhecimento que está adquirindo com o conhecimento que já possui.

Aplicar: esse nível está relacionado a capacidade do aluno em executar um determinado procedimento em uma determinada situação.

Analisar: esse nível está relacionado a capacidade do aluno em conseguir identificar partes importantes e não tão importantes sobre uma informação do conteúdo, conseguindo fazer um relação entre essas partes.

Avaliar: esse nível está relacionado a capacidade do aluno em julgar um determinado problemas, a fim de reconhecer a estratégia de resolução.

Criar: esse nível está relacionado a capacidade do aluno em criar uma nova solução para os problemas pelos quais foi desafiado, a partir das habilidades adquiridas.

Apesar de haver uma predisposição dos pesquisadores em adotar a Taxonomia de Bloom para apoiar a definição dos objetivos de aprendizagem, não existe uma só forma de escrever os objetivos de aprendizagem. Obviamente, a utilização de alguma taxonomia pode auxiliar na redação dos objetivos. Em especial, a Taxonomia de Bloom auxilia esse processo de escrita porque possui um verbo associado a cada nível do domínio cognitivo.

Considerando que há estudos que se inspiram nessa taxonomia para definir objetivos de aprendizagem de conteúdos (KIWELEKAR; WANKHEDE, 2015), para definir os objetivos de aprendizagem para cada conteúdo contemplado pela abordagem FlipClass-STT, os níveis da dimensão cognitiva de Bloom foram apreciados. Apesar da inspiração, a pesquisa não se comprometeu em abordar todos os verbos que caracterizam os níveis, uma vez que o conteúdo possui especificidades que não são triviais de ser classificadas nos níveis.

Nessa perspectiva, os objetivos de aprendizagem foram definidos, inspirados na Taxonomia de Bloom, e foram estabelecidos para guiar o professor que utilizará a abordagem. Assim, para definir os objetivos, foi necessário escrever objetivos de aprendizagem para cada uma das técnicas e critério de teste de software, pré-definidos na seção anterior. A Tabela 26 apresenta a lista de objetivos de aprendizagem definidos para a abordagem.

Salienta-se que a elaboração da abordagem é por si só um objeto de pesquisa complexo. Assim, ressalta-se que a Taxonomia de Bloom não é explorada com profundidade, sendo usada apenas como um instrumento para apoiar a redação dos objetivos. Pesquisas mais aprofundadas precisam ser feitas para escrever com maior precisão a redação dos objetivos de aprendizagem. Apesar disso, acredita-se que os objetivos estabelecidos cumprem com o que é esperado pela abordagem, uma vez que buscam facilitar o planejamento do processo de ensino e aprendizagem.

Tabela 26 – Os objetivos de aprendizagem da abordagem FlipClass-STT

Identificador	Tópico	Objetivo de Aprendizagem	Categoria cognitiva
#1	Critérios baseados em fluxo de controle	Identificar a fonte de informação usada para a condução da atividade de teste.	Recordar
#2		Entender e saber gerar a partir do código o grafo de fluxo de controle (GFC)	Compreender
#3		Conhecer os critérios da família de critérios baseados em fluxo de controle, em especial: Todas-Arestas e Todos-Nós.	Compreender
#4		Aplicar os critérios Todas-Arestas e Todos-Nós na condução da atividade de teste, considerando o uso de uma pequena unidade de software.	Aplicar
#5	Critérios baseados em fluxo de dados	Identificar a fonte de informação usada para a condução da atividade de teste.	Recordar
#6		Entender e saber gerar a partir da informação o grafo definição-uso	Compreender
#7		Conhecer os critérios da família de critérios baseados em fluxo de dados, em especial: Todas-Definições, Todos-Usos, Todos-Potencias-Usos.	Compreender
#8		Aplicar os critérios baseados em fluxo de dados na condução da atividade de teste, utilizando pequenas unidades de software (e.g., algoritmos);	Aplicar
#9	Teste de mutação	Identificar a fonte de informação usada para a condução da atividade de teste.	Recordar
#10		Conhecer o critério teste de mutação	Compreender
#11		Compreender o processo de aplicação do critério de teste de mutação	Compreender
#12		Aplicar o critério teste de mutação em programas implementados na linguagem C.	Aplicar
#13	Teste funcional	Caracterizar a técnica de teste funcional	Recordar
#14		Reconhecer as principais características da técnica, em especial a fonte de informação utilizada para aplicação dos critérios	Compreender
#15		Aplicar os critérios de particionamento em classes de equivalência e análise de valor limite em pequenas unidades de um software.	Aplicar
#16		Elaborar um conjunto otimizado de casos de teste para satisfazer os critérios, tendo como base uma especificação de um software.	Aplicar

Fonte: Elaborada pelo autor.

4.2.3 Definição de recursos educacionais

A definição dos recursos educacionais deve obrigatoriamente considerar o conteúdo e os objetivos de aprendizagem. Ela também pode respeitar o perfil da turma em que o modelo será aplicado. Para apoiar a identificação do perfil da turma e posteriormente definir quais os materiais didáticos que são indicados, alguns pesquisadores da área sugerem a detecção dos estilos de aprendizagem dos alunos (FASSBINDER *et al.*, 2014). Estilos de aprendizagem remetem à forma como o estudante prefere receber e processar

a informação (FELDER; SILVERMAN, 1998).

De acordo com Lyra *et al.* (2016) alguns estudantes, por exemplo, preferem dados e informações visuais enquanto outros optam por teorias e informações verbais. Nessa perspectiva, os estilos de aprendizagem dos alunos fazem com que eles fiquem confortáveis ao receber e processar uma informação. Por esse motivo, há uma frente de estudos que realça que os professores devem identificar os estilos de aprendizagem de seus estudantes, em prol de uma aprendizagem efetiva (NAPS *et al.*, 2003).

Apesar de existir uma predisposição dos autores em recomendar a identificação dos estilos de aprendizagem para a seleção e definição dos recursos educacionais em aplicações do modelo *flipped classroom*, como é o caso do estudo de Fassbinder *et al.* (2014) que recomenda o modelo de Felder-Silverman e o instrumento ILS, existem algumas vertentes contrárias a estilos de aprendizagem. De acordo com Kirschner (2017), a validade e confiabilidade dos instrumentos que utilizados para detectar os estilos é questionável.

Baseando-se nisso, durante a definição da abordagem, foi estabelecido que a identificação do estilo de aprendizagem não seria considerada. Assim, na sequência, recursos educacionais desenvolvidos por terceiros foram analisados. Para apoiar a análise e seleção, o mapeamento sistemático conduzido por Valle, Barbosa e Maldonado (2015) foi recorrido. Nesse estudo, os autores fazem um levantamento das abordagens desenvolvidas para apoiar o ensino de teste de software. Dentre elas, os autores destacam os jogos educacionais e os módulos educacionais.

Dentre os jogos existentes que oferecem subsídios ao ensino de teste de software podem ser mencionados: U-TEST (THIRY; ZOUCAS; SILVA, 2011), Jogo das 7 Falhas (DINIZ; DAZZI, 2011), JETS (SILVA, 2012), TestEG (OLIVEIRA, 2013), ItestLearning (FARIAS *et al.*, 2012), JoVeTest (BARBOSA *et al.*, 2016), Code Defenders (ROJAS *et al.*, 2017) e Testing Game (VALLE, 2017). Cada um desses jogos possui características específicas no que diz respeito ao ensino do planejamento do teste, técnicas e critérios de teste.

A disponibilidade do jogo foi analisada. Foi possível reconhecer que apenas três jogos estão atualmente disponíveis, são eles: Code Defenders, JoVeTest e Testing Game. O jogo Code Defenders pode ser acesso e jogado por intermédio de navegadores *web*, através do seguinte endereço: <http://code-defenders.org/>. JoVeTest tem como requisito para sua execução um versão igual ou superior ao Java 8 e está disponível em <http://experts.icomp.ufam.edu.br/jovetest/ >. Finalmente, Testing Game ainda não foi hospedado em um servidor para acesso, mas o seu código-fonte está disponível em: <https://bitbucket.org/pedrohdvalle/testinggamehtml5>.

Foi constatado que todos os jogos requisitam conhecimento prévio dos estudantes no que se refere ao conteúdo abordado no jogo. Nesse sentido, considerando os diferentes

momentos previstos pelo modelo *flipped classroom*, eles são mais indicados para uso no momento "depois da aula". Além disso, a maioria dos jogos não estão disponíveis, o que inviabiliza o uso deles na abordagem FlipClass-STT. Diante disso, após a análise, os módulos educacionais, revelados no mapeamento de Valle, Barbosa e Maldonado (2015), foram revisados.

Barbosa (2004) construiu módulos para o domínio de teste de software. Para tanto, a autora estruturou, modelou e implementou os conceitos, fatos, princípios, procedimentos, exemplos, informações complementares e exercícios pertinentes ao contexto de teste de software, integrando páginas HTML (*HyperText Markup Language*), documentos-texto e ferramentas específicas do domínio de conhecimento, em especial as ferramentas para apoiar a automatização da atividade: Poke-Tool (CHAIM, 1991), a versão *web* da Proteum 2.0 (DELAMARO; MALDONADO, 2001) e a JaBUTi (VINCENZI, 2004). Ao total, foram especificados 16 módulos para o domínio de teste de software.

Os módulos de Barbosa (2004) contemplam aspectos instrucionais que podem ser utilizados para preparar os estudantes para a aula, no momento "antes da aula". Todavia, eles não estão liberados para acesso e uso, não estão disponíveis em domínio público ou possuem licença aberta, inviabilizando a seleção dos mesmos para apoiar a abordagem. Diante disso, foi necessário recorrer a outras iniciativas da área de teste de software que buscam apoiar o ensino.

Seguindo a sugestão de Bergmann e Sams (2012) que descrevem sobre a possibilidade de utilização de vídeos instrucionais produzidos por terceiros, bem como as recomendações de Fassbinder *et al.* (2014) que descrevem sobre materiais didáticos produzidos e licenciados sob licença aberta em cursos massivos *online* (MOOCs), alguns vídeos educacionais disponíveis em MOOCs foram analisados.

Análise de Vídeos Educacionais

Recorreu-se aos vídeos instrucionais disponibilizados em plataformas MOOC, uma vez que estudos mencionam que os vídeos podem ser alternativas interessantes para aplicações do modelo *flipped classroom*, porque precisam possuir uma qualidade que atenda os critérios estabelecidos pela plataforma MOOC (FASSBINDER *et al.*, 2014). Além disso, são vídeos com duração menor, em que um tópico é equivalente a um vídeo, em conformidade com o que é recomendado por Bergmann e Sams (2012).

Diante disso, foram localizados cursos MOOC sobre teste de software que possuíam o conteúdo de teste funcional. Foram encontrados cursos nos idiomas português e inglês, oferecidos pelas plataformas UDACITY, edX, Alison e Coursera. Esses cursos, em geral foram criados por professores e/ou pesquisadores de renomadas instituições de ensino (SILVA; BARBOSA; FASSBINDER, 2017).

Durante a análise foi observado o conteúdo dos vídeos, se eles apresentam exemplos práticos, recomendação de exercícios, o tempo de duração dos vídeos, as suas resoluções, o idioma de cada vídeo, a disponibilidade de legenda, e, por fim, se eles possuem uma opção para download. Cada um desses itens foi definido com base na sua significância para contribuir para a abordagem. O item "conteúdo", por exemplo, apoiou a identificação do conteúdo abordado no vídeo. Esse item não reflete ao título do vídeo, uma vez que alguns vídeos apresentam títulos mais genéricos, e não indicam, por exemplo, os critérios que são abordados.

O item "exemplo" contribuiu com o reconhecimento sobre como o conteúdo foi abordado, isto é, se o vídeo contém um ou mais exemplos. Esse item foi definido especialmente porque era esperado que os vídeos sobre critérios de teste abordassem exemplos de aplicações dos critérios, não somente a teoria que os fundamentam. O item "exercícios", por outro lado, ofereceu suporte a detecção de indicações de exercícios que são feitas durante o vídeo. Nesse sentido, foi verificado se o vídeo recomenda exercícios para que o aluno possa fazê-los durante a instrução ou ao final dela.

Outro item observado foi o tempo de duração do vídeo, uma vez que os vídeos recomendados aos alunos não devem ser extremamente longos. Espera-se que os vídeos não sejam muito extensos, pois os alunos precisarão assisti-los completamente, inclusive algumas vezes assistir novamente partes do vídeo que não ficaram claras em um primeiro momento. Além disso, os alunos terão que se preparar para aula e gerenciar atividades de outras disciplinas que eles estão matriculados e cursando em paralelo.

O idioma do vídeo e a opção de legenda foram outras características analisadas. Dado que o público-alvo da abordagem é constituído, principalmente, por professores de computação do Brasil, foi definido que o idioma do vídeo é um fator importante. Em um primeiro momento, foi ponderado que o idioma poderia ser o inglês e/ou o espanhol, mas o vídeo deveria oferecer a opção de legenda em português.

Aspectos envolvidos na legibilidade dos textos e caracteres utilizados nos vídeos educacionais são considerados relevantes durante a seleção dos vídeos (SCHNEIDER; CAETANO; RIBEIRO, 2012). Nesse sentido, um item que reflete na legibilidade é a resolução do vídeo. Uma resolução muito baixa, por exemplo, pode influenciar na visualização do vídeo.

Por fim, foi averiguado se os vídeos oferecem a opção de download para o aluno. Esse último item foi considerado importante, uma vez que no primeiro momento previsto pelo *flipped classroom* (*i.e.*, "antes da aula"), os estudantes precisarão acessar os vídeos em suas casas. Como o acesso pode ficar limitado ao tipo de conexão com a Internet que o aluno possui em sua casa, oferecer uma opção para download pode flexibilizar as opções de acesso ao material.

Uma visão geral sobre os vídeos analisados é apresentada na Tabela 27. Nela é possível observar a origem deles e o tempo total que cada vídeo possui. Salienta-se que o tempo se remete apenas ao conteúdo que é compreendido pela abordagem FlipClass-STT. Esses cursos oferecidos abordam outros temas associados a temática de teste de software, como qualidade de software, gerenciamento da atividade, verificação e validação de especificação, dentre outros. Além disso, alguns MOOC abordam conteúdos relacionados a outras abordagens de teste, como teste de *stress*, teste de desempenho, teste de GUI (*Graphical User Interface*) e teste de usabilidade.

Diante das opções de vídeo encontradas e da análise realizada, foi possível observar que os vídeos com menor qualidade visual foram os disponibilizados na plataforma Coursera (apenas 480p). Contudo, esse material é o único que menciona a técnica de teste baseado em defeitos e o critério teste de mutação. Nessa perspectiva, foram selecionados os vídeos do MOOC Coursera em virtude dos mesmos possuírem uma visão teórico-prática sobre o critério de mutantes.

Tabela 27 – Descrição sobre os cursos MOOC sobre teste de software

Origem	**Entidade**	**Nome do curso**	**Técnicas de teste**			**Total**
			Teste funcional	**Teste estrutural**	**Teste baseado em defeitos**	
			Tempo total de instrução			
Udacity	Amazon Web Services	Software Testing	5 minutos e 15 segundos	10 minutos e 18 segundos	0 minuto	15 minutos e 33 segundos
edX	Delft University of Technology	Automated Software Testing: Practical Skills for Java Developers	35 minutos e 54 segundos	16 minutos e 44 segundos	0 minuto	52 minutos e 38 segundos
Alison	National Programme on Technology Enhanced Learning	Introduction to Software Testing	25 minutos e 43 segundos	0 minuto	0 minuto	25 minutos e 43 segundos
Coursera	Universidade de São Paulo	Introdução ao Teste de Software	53 minutos e 12 segundos	27 minutos e 7 segundos	21 minutos e 58 segundos	102 minutos e 17 segundos

Fonte: Elaborada pelo autor.

Adicionalmente, outras características foram apreciadas, a saber: (i) cada conteúdo deve possuir um vídeo sobre a descrição teórica do assunto, um vídeo com exemplos práticos de aplicação dos critérios de teste e descrição sobre ferramentas de apoio a automatização do critério; (ii) o vídeo deve conter instruções que atendem objetivos de aprendizagem associados aos dois primeiros níveis da Taxonomia de Bloom. A Tabela 28 apresenta uma análise sobre os vídeos da plataforma Coursera. A análise completa a respeito dos 37 vídeos investigados está disponível em <http://bit.ly/2TJOzgy>.

Tabela 28 – Análise dos vídeos instrucionais sobre teste de software

Plataforma	Identificador	Conteúdo	Exemplos	Exercícios	Tempo de duração	Idioma do vídeo	Legendas	Resolução do vídeo	Download
Coursera	v25	Fases da Atividade de Teste	Não	Não	8 minutos e 37 segundos	Português	Português	480p	Disponível
	v26	Técnicas e critérios de teste	Não	Não	8 minutos e 19 segundos	Português	Português	480p	Disponível
	v27	Contextualizando a técnica teste funcional	Não	Não	5 minutos e 58 segundos	Português	Português	480p	Disponível
	v28	Critério particionamento em classes de equivalência	Sim	Sim	9 minutos e 51 segundos	Português	Português	480p	Disponível
	v29	Análise do valor limite	Sim	Sim	8 minutos e 23 segundos	Português	Português	480p	Disponível
	v30	Um panorama das ferramentas de apoio ao teste funcional	Não	Não	2 minutos e 46 segundos	Português	Português	480p	Disponível
	v31	Teste funcional e JUnit	Sim	Não	7 minutos e 18 segundos	Português	Português	480p	Disponível
	v32	Demonstração de JUnit em Teste Funcional	Sim	Não	18 minutos e 56 segundos	Português	Português	480p	Disponível
	v33	Contexto geral do Teste Estrutural	Sim	Sim	8 minutos	Português	Português	480p	Disponível
	v34	Critérios de fluxo de controle (Todos-Nós e Todos-Arcos) e fluxo de dados (Definição-Uso)	Sim	Sim	10 minutos e 20 segundos	Português	Português	480p	Disponível
	v35	Critérios baseados em fluxo de controle e exemplo de aplicação	Sim	Sim	8 minutos e 47 segundos	Português	Português	480p	Disponível
	v36	Contexto geral sobre o teste de mutação	Sim	Sim	10 minutos e 30 segundos	Português	Português	480p	Disponível
	v37	Uma visão prática sobre o teste de mutação	Sim	Sim	11 minutos e 28 segundos	Português	Português	480p	Disponível

Fonte: Elaborada pelo autor.

Após a definição dos vídeos instrucionais disponíveis no curso da plataforma Coursera, observou-se que os exemplos e exercícios discutidos nesses vídeos poderiam ser reaproveitados para apoiar as atividades realizadas no momento "durante a aula". Apesar dos vídeos abrangerem o conteúdo da abordagem, outros mecanismos foram investigados. Assim, visando reforçar o conteúdo que compreende FlipClass-STT, foram analisados materiais produzidos por professores autores dos vídeos do curso "Introdução ao Teste de Software". Nesse sentido, foi possível observar que os autores desses vídeos possuem um conjunto de recursos educacionais abertos desenvolvidos para o ensino de teste de software, disponíveis em: <http://napsol.icmc.usp.br/ats/>.

Análise de Recursos Educacionais Abertos

O material produzido pelos autores do MOOC intitulado "Introdução ao Teste de Software" em conjunto com outros pesquisadores da área de teste de software, foi desenvolvido no âmbito do Centro de Competência em Software Livre do Instituto de Ciências Matemática e de Computação da Universidade de São Paulo (CCSL-USP), sob coordenação do professor Dr. José Carlos Maldonado. Esse centro integra a rede internacional de Centros de Competência do Projeto QualiPSo (*Quality Platform for Open Source Software*[6]).

[6] Mais informações podem ser obtidas em: <http://bit.ly/2SuhNDN>.

Esses materiais contêm conteúdo da área de teste de software, em formato de tutoriais e slides. Eles foram produzidos para um curso sobre automatização da atividade de teste de software. Por consequência, aborda ferramentas de teste com licença livre (*open source*). A licença dos materiais que foi adotada pelos autores permite o compartilhamento (copiar e distribuir o material em qualquer suporte ou formato) e a adaptação (transformar e criar a partir do material) e não permite o uso do conjunto de materiais para fins comerciais. Nessa perspectiva, esses mecanismos de apoio ao ensino de teste são classificados como recursos educacionais abertos.

Recursos Educacionais Abertos (REA) são materiais especializados para ensinar, aprender e pesquisar, que estão em domínio público ou são publicados com licença de propriedade intelectual que possibilita o seu livre acesso, adaptação e distribuição (AMIEL; OREY; WEST, 2010; ARIMOTO; BARBOSA, 2013; BUTCHER, 2015). Esses recursos surgiram com o aflorar das Tecnologias de Informação e Comunicação Aplicadas na Educação e permitem que professores, estudantes e demais profissionais da educação possam copiar, adaptar e compartilhar seus recursos educacionais de forma livre.

Uma análise sobre o conteúdo dos REA foi realizada, visando investigar as possibilidades do material como recursos educacionais para a abordagem FlipClass-STT. Nessa análise, observou-se o formato do REA, o conteúdo abordado, se ele contém exemplo, recomenda exercícios e o seu número de páginas. O formato do recurso educacional foi usado para indicar ao utilizador da abordagem qual o tipo de recurso disponível. Notou-se que todos os REA tem como característica o formato de slides, baseados em recursos multimídia e hipertexto.

Nos REA, foi possível identificar algumas atividades que podem ser exploradas pelo professor que adotará a abordagem FlipClass-STT. Foram reconhecidas ferramentas que apoiam o ensino sobre a automatização dos critérios de teste. O número de páginas dos REA foi observado, em especial, notou-se que os REA abordam detalhadamente o conteúdo, principalmente os REA que abordam as técnicas de teste e seus critérios, os quais em todo caso apresentam exemplos e recomendam exercícios.

A Tabela 29 apresenta a análise completa. Salienta-se que existem outros REA sobre o conteúdo de teste de software produzidos pelos autores do CCSL-USP. Entretanto, a análise contempla apenas o conteúdo associado ao nível de teste, técnicas e critérios de teste, em virtude do conteúdo definido na Seção 4.2.1.

Após a análise dos REA, foi definido que eles complementariam os vídeos instrucionais selecionados e descritos na seção anterior, especialmente os vídeos relacionados aos critérios de fluxo de controle e fluxo de dados. Esses recursos também foram selecionados para apoiar a definição das atividades que compõe a abordagem.

Tabela 29 – Análise dos recursos educacionais abertos sobre teste de software

Identificador	Formato do recurso	Conteúdo	Exemplos	Exercícios	Número de páginas
REA1	Slide	Conceitos sobre teste de software e fases de teste (teste de unidade, teste de integração, teste de sistema, teste de aceitação, teste alfa e beta)	Sim	Não	104
REA2	Slide	Framework para teste de unidade (JUnit)	Sim	Não	100
REA3	Slide	Uma visão geral sobre a técnica de teste funcional	Não	Não	9
REA4	Slide	Critério Particionamento de Classes Equivalentes	Sim	Sim	35
REA5	Slide	Exercício sobre o Critério Particionamento de Classes Equivalentes	Não	Sim	12
REA6	Slide	Critério Análise do Valor Limite	Sim	Sim	24
REA7	Slide	Uma visão geral sobre a técnica de teste estrutural	Não	Não	18
REA8	Slide	Critérios baseados em fluxo de controle (grafo de fluxo de controle, complexidade ciclomática, todos-nós, todos-arcos, todos-caminhos, teste do caminho básico)	Sim	Sim	60
REA9	Slide	Guia de instalação da ferramenta EclEmma 2.2.1 : um software de apoio ao teste estrutural	Sim	Sim	53
REA10	Slide	Guia de instalação da ferramenta Emma 2.1 : um software de apoio ao teste estrutural	Sim	Sim	26
REA11	Slide	Critérios de fluxo de dados (todas as definições, todos os p-usos, todos os p-usos e alguns c-usos, todos os c-usos e alguns p-usos, todos os usos, potenciais-usos)	Sim	Sim	50
REA12	Slide	Ferramenta JaBUTI	Sim	Sim	114
REA13	Slide	Uma visão geral sobre o teste baseado em defeitos, teste de mutação, mutantes de interface e a ferramenta Proteum	Sim	Sim	78
REA14	Slide	Ferramenta MuJava: um software de apoio ao teste de mutação	Sim	Não	19
REA15	Slide	Combinando critérios de teste de software (particionamento em classes de equivalência, todos-arcos, todos-potenciais-usos, análise de mutante e mutação de interface)	Sim	Não	78

Fonte: Elaborada pelo autor.

Por fim, para complementar a análise de recursos educacionais para apoiar a abordagem, foram analisados objetos de aprendizagem produzidos por terceiros, os quais foram socializados recentemente por pesquisadores da área de ensino de teste de software e não foram revelados na pesquisa de Valle, Barbosa e Maldonado (2015). A existência dos objetos de aprendizagem foi reconhecida durante a revisão de literatura desta dissertação.

Objetos de Aprendizagem

Objetos de aprendizagem são recursos digitais reusáveis formatados por unidades de conteúdos (POLSANI, 2003; GORDILLO; BARRA; QUEMADA, 2017). Essas entidades são constituídas por elementos independentes que utilizam um conjunto de metadados para possibilitar que as mesmas sejam indexadas e recuperadas por repositórios (SILVA, 2011). Apesar de não existir um senso comum para contextualizar um objeto de aprendizagem, muitas pesquisas referenciam a elucidação da IEEE LTSC (*Learning Technology Standards Committee*) que os define como "qualquer entidade, digital ou não, que pode ser usada para o ensino, educação ou treinamento".

No estudo de Benitti (2015) é mencionado o desenvolvimento de objetos de aprendizagem para o ensino de teste de software. Ao total foram planejados 18 objetos de aprendizagem e até o presente momento foram desenvolvidos e disponibilizados 15 objetos de aprendizagem. Oito objetos remetem as técnicas e critérios de teste, quatro são

destinados a detalhamento sobre níveis de teste, um aborda os aspectos conceituais da atividade e dois focam nos processos de teste.

Visando aproveitar os benefícios educacionais dos objetos de aprendizagem, eles também foram analisados. Dos 15 objetos, foram considerados apenas nove, os quais contemplam as técnicas de teste funcional, estrutural e baseada em defeitos. Para apoiar a analise, foram definidos alguns itens que seriam verificados, são eles: conteúdo, exemplo, exercício e formato do objeto de aprendizagem. Para o contexto de objetos de aprendizagem, o formato do mesmo é importante, uma vez que precisam utilizar padrões que favorecem a aprendizagem eletrônica (*e-learning*).

A Tabela 30 apresenta a análise completa. Nota-se que todos os objetos são constituídos por exercícios. Esses exercícios têm características teórico-prática e são realizados no próprio objeto de aprendizagem. Ao final do exercício, o objeto de aprendizagem apresenta um *feedback* sobre a atividade realizada, indicando ao aluno se ele conseguiu responder corretamente a atividade ou não. Todos os objetos podem ser acessados por *browsers* porque foram desenvolvidos em HTML5. Além disso, os autores construíram eles no padrão SCORM (*Sharable Content Object Reference Model*), um modelo de referência que recomenda como o objeto deve ser organizado, como o conteúdo deve ser estruturado e como ele será publicado (TAROUCO; SCHMITT, 2010).

Os objetos de aprendizagem mencionados na Tabela 30 foram definidos para compor os recursos educacionais da abordagem. Eles também apoiaram a identificação de atividades que podem ser utilizadas pelo professor que irá adotar a abordagem. Durante a recomendação do material, visando instigar os alunos a solucionarem as atividades disponíveis nos objetos, o professor pode recomendar a utilização dos objetos de aprendizagem. Nessa perspectiva, os objetos de aprendizagem podem ser usados tanto para o treinamento quanto para a fixação do conteúdo.

4.2.4 Definição de atividades

Após a análise de recursos educacionais para definir o conteúdo que faz parte da abordagem, foi necessário determinar as atividades que constituem a abordagem. Em um primeiro momento, foram adotadas as atividades recomendadas nos REA desenvolvidos pelos membros do CCSL-USP. Elas envolvem a prática da aplicação dos critérios de teste, atendendo os objetivos de aprendizagem definidos na abordagem. Também, foram selecionadas atividades que haviam sido usadas nos estudos experimentais desta dissertação, descritas no Capítulo 3. Como o intuito de complementar as atividades, foram selecionados exemplos disponíveis na literatura da área, em especial os exemplos descritos em Myers, Sandler e Badgett (2011), Delamaro, Maldonado e Jino (2016) e Maldonado, Delamaro e Vincenzi (2018).

Tabela 30 – Análise dos objetos de aprendizagem sobre teste de software

Identificador	Conteúdo	Exemplo	Exercício	Formato	Local
OA1	Introdução as técnicas de teste de software, conceitualização sobre teste estrutural e teste funcional	Não	Sim	SCORM / HTML5	http://bit.ly/2GHagdq
OA2	Teste estrutural Grafo de Fluxo de Controle, Todos-Nós, Todos-Arcos, Todos-Caminhos, complexidade ciclomática	Sim	Sim	SCORM / HTML5	http://bit.ly/2N3GIIa
OA3	Exemplos de aplicação de critérios baseados em fluxo de controle (todos-nós, todas-arestas, todas-condições e todos-caminhos)	Sim	Sim	SCORM / HTML5	http://bit.ly/2DAKPqL
OA4	Exemplos de aplicações de critérios baseados em fluxo de dados (todas-definições, todos-usos)	Sim	Sim	SCORM / HTML5	http://bit.ly/2TOVW6r
OA5	Exemplos de aplicações de critérios baseados na complexidade (Todos-caminhos)	Sim	Sim	SCORM / HTML5	http://bit.ly/2DApH4b
OA6	Exemplos de aplicações de critérios funcionais (particionamento em classes de equivalência e análise do valor limite)	Sim	Sim	SCORM / HTML5	http://bit.ly/2GLvBlU
OA7	Teste de unidade	Sim	Sim	SCORM / HTML5	http://bit.ly/2SzHVNm
OA8	Teste de integração	Sim	Sim	SCORM / HTML5	http://bit.ly/2BA4Wp3
OA9	Teste de sistema e aceitação	Sim	Sim	SCORM / HTML5	http://bit.ly/2V0JfFZ

Fonte: Elaborada pelo autor.

4.3 FlipClass-STT: uma visão geral

Durante a seção anterior, foram descritos procedimentos conduzidos para apoiar a definição da abordagem FlipClass-STT, que por sua vez busca subsidiar a implementação do modelo. Assim, foi necessário selecionar o conteúdo, definir objetivos de aprendizagem para o conteúdo, investigar recursos educacionais e atividades para estabelecer um conjunto de materiais didáticos que contribuem para que os objetivos de aprendizagem sejam alcançados. Com base nisso, foi possível determinar o conjunto de mecanismo de apoio à aplicação de *flipped classroom*.

A tarefa associada à definição dos mecanismos de apoio (materiais didáticos e atividades) contribui para intensificar o trabalho do professor que tem a intenção de implementar o modelo *flipped classroom* (GANNOD; BURGE; HELMICK, 2008). Assim, todas as atividades descritas na seção que antecede a atual foram cruciais para a definição da abordagem FlipClass-STT. Consequentemente, contribuem para o planejamento de uma aula de teste de software baseada no modelo, minimizado o esforço de um professor que está interessado em utilizá-lo.

Para organizar o planejamento de forma mais efetiva, o *template* estruturado por Gilboy, Heinerichs e Pazzaglia (2015) foi usado. Ele foi apresentado na Figura 3, no Capítulo 2. Esse *template* busca metodizar principalmente os momentos previstos pelo modelo e as atividades que devem ser realizadas pelos estudantes em cada um dos momentos. Ele é constituído por cinco itens/campos, a saber: nome do tópico, objetivos de aprendizagem para o tópico, recursos, atividades e avaliação.

Nome do tópico: Nesse item é indicado o conteúdo que é abordado no planejamento

(*e.g.*, teste funcional -– critérios particionamento em classes de equivalência e análise do valor limite). Para o contexto da abordagem FlipClass-STT, esse campo apresenta o tópico definido a partir do *survey*.

Objetivos de aprendizagem: Nesse campo, são relatados os objetivos de aprendizagem estabelecidos para ensinar o conteúdo de teste. No âmbito da abordagem FlipClass-STT, eles foram redigidos com base na Taxonomia de Bloom e experiência do autor na área, sendo revisados pela orientadora da pesquisa.

Recursos: No campo recursos são listados os recursos educacionais ou materiais didáticos que serão recomendados aos estudantes. Para o contexto desta pesquisa, a abordagem FlipClass-STT aborda um conjunto de recursos educacionais produzidos por terceiros e oferecidos em formato de "conteúdo aberto". Nesse sentido, compreende os vídeos instrucionais de MOOC (mencionados na Tabela 28), REA (listados na Tabela 29) e objetos de aprendizagem (elencados na Tabela 30).

Atividades: No *template* há um campo específico para as atividades que os alunos devem realizar durante o aprendizado. Esse campo contempla os três momentos previstos pelo modelo. Essas atividades não refere-se somente a execução de exercícios práticos, mas tarefas de precisam ser realizadas pelos estudantes, como acessar o material didático, assistir os vídeos, dentre outras. Para o contexto da abordagem, foram definidas atividades prévias que recomendam ao aluno um conjunto de ações que ele precisa realizar, bem como a sequência exata de cada ação.

Avaliação: O último item compreende a avaliação, em que é descrito como o aluno será avaliado durante a implementação de *flipped classroom*. Semelhantemente ao item "atividades", a avaliação compreende todas os momentos previstos pelo modelo. A abordagem FlipClass-STT recomenda que o professor avalie o aluno durante a aula, uma vez que nesse momento os alunos já terão passado por uma série de atividades instrucionais (*i.e.*, estudo sobre o conteúdo, resolução de dúvidas, trabalho em equipe).

Com base em toda as atividades realizadas e definições tomadas, a abordagem foi idealizada. Ela é constituída pelo planejamento de implementação de *flipped classroom* para os conteúdos de teste de software. A Figura 23 contém o planejamento para ensino de teste funcional e os critérios Particionamento em Classes de Equivalência e Análise do Valor Limite. A Figura 24 ilustra o planejamento para implantação de *flipped classroom* para os critérios baseados em fluxo de controle, mais especificamente os critérios Todos-Nós e Todas-Arestas. Na Figura 25 é relatado o planejamento para o ensino dos critérios baseados em fluxo de dados, especialmente Todas-Definições, Todos-Usos, Todos-Potenciais-Usos. A Figura 26 inclui o planejamento associado a técnica de teste baseado em defeitos e o critério teste de mutação.

Figura 23 – Planejamento da implementação de Flipped Classroom para Teste Funcional

Nome do Tópico: Teste Funcional e os critérios Particionamento em Classes de Equivalência e Análise do Valor Limite

Objetivos de aprendizagem para o tópico:
- Caracterizar a técnica de teste funcional;
- Reconhecer as principais características da técnica, em especial a fonte de informação utilizada para aplicação dos critérios;
- Aplicar os critérios de Particionamento em Classes de Equivalência e Análise de Valor Limite em diferentes sistemas de software;
- Elaborar um conjunto otimizado de casos de teste para satisfazer os critérios, tendo como base uma especificação de um software

Recursos:
- Vídeos instrucionais
  - Teste Funcional: contexto e funcionamento. Disponível em: <http://bit.ly/2GFZC6M>
  - Critério: Particionamento em Classes de Equivalência. Disponível em: <http://bit.ly/2X3R1AE>
  - Critério: Análise do Valor Limite. Disponível em: <http://bit.ly/2BzP8lW>

	Antes da aula	**Durante a aula**	**Depois da aula**
Atividades	1. Acessar os mecanismos de apoio. 2. Eliminar as distrações (*e.g.*, sair das redes sociais). 3. Assistir os vídeos (utilizar a vontade recursos de Pausar e Retroceder do player) . 4. Fazer anotações sobre o conteúdo. 5. Refazer os exemplos disponibilizados. 6. Anotar as dúvidas.	1. Solucionar as dúvidas que surgiram antes da aula. 2. Realizar a atividade de fixação (o professor irá orientar o aluno a respeito da atividade). Atividade sugerida disponível em: <http://bit.ly/2DHSvaX>.	1. Acessar o objeto de aprendizagem listado a seguir para reforçar o conteúdo aprendido. Objeto disponível em: <http://bit.ly/2GLvBlU>. 2. Estudar o conteúdo da próximo aula.
Avaliação	Não se aplica	Atividade prática	Não se aplica

Fonte: Elaborada pelo autor.

Figura 24 – Planejamento da implementação de Flipped Classroom para Critérios baseados em Fluxo de Controle

Nome do Tópico: Teste Estrutural e os principais critérios Baseados em Fluxo de Controle

Objetivos de aprendizagem para o tópico:
- Identificar a fonte de informação usada para a condução da atividade de teste;
- Entender e saber gerar a partir do código o grafo de fluxo de controle (GFC);
- Conhecer os critérios da família de critérios baseados em fluxo de controle, em especial: Todas-Arestas e Todos-Nós;
- Aplicar os critérios Todas-Arestas e Todos-Nós na condução da atividade de teste, considerando o uso de uma pequena unidade de software.

Recursos:
- Vídeos instrucionais
  - Introdução ao Teste Estrutural. Disponível em: <http://bit.ly/2P4ImO6>
  - Critérios de Fluxo de Controle. Disponível em: http://bit.ly/2DgYWDT
- Objeto de aprendizagem
  - Os critérios baseados no fluxo de controle. Disponível em: <http://bit.ly/2N3GIIa>

	Antes da aula	**Durante a aula**	**Depois da aula**
Atividades	1. Acessar os mecanismos de apoio. 2. Eliminar as distrações (e.g. sair das redes sociais). 3. Assistir os vídeos (utilizar a vontade recursos de Pausar e Retroceder do player) . 4. Fazer anotações sobre o conteúdo. 5. Refazer os exemplos disponibilizados. 6. Anotar as dúvidas.	1. Solucionar as dúvidas que surgiram antes da aula. 2. Realizar a atividade de fixação (o professor irá orientar o aluno a respeito da atividade). Atividade sugerida disponível em: <http://bit.ly/2TNQL6F>. 3. Fazer uma atividade prática com ferramenta. Atividade sugerida disponível em: <http://bit.ly/2UTAZHu>.	1. Acessar o objeto de aprendizagem listado a seguir para reforçar o conteúdo aprendido. Objeto disponível em: <http://bit.ly/2DAKPqL>. 2. Estudar o conteúdo da próximo aula.
Avaliação	Não se aplica	Atividade prática	Não se aplica

Fonte: Elaborada pelo autor.

Figura 25 – Planejamento da implementação de Flipped Classroom para Critérios baseados em Fluxo de Dados

Nome do Tópico: Teste Estrutural e os principais critérios Baseados em Fluxo de Dados

Objetivos de aprendizagem para o tópico:
- Identificar a fonte de informação usada para condução da atividade;
- Entender e saber gerar a partir da de informação o grafo definição-uso;
- Conhecer os critérios da família de critérios baseados em fluxo de dados, em especial: Todas-Definições, Todos-Usos, Todos-Potenciais-Usos;
- Aplicar os critérios baseados em fluxo de dados na condução da atividade de teste, utilizando pequenas unidades de software (*e.g.*, algoritmos);

Recursos:
- Objeto de aprendizagem
  - Os critérios baseados no fluxo de dados. Disponível em: <http://bit.ly/2TOVW6r>.

	Antes da aula	**Durante a aula**	**Depois da aula**
Atividades	1. Acessar os mecanismos de apoio. 2. Eliminar as distrações (*e.g.*, sair das redes sociais). 3. Assistir os vídeos (utilizar a vontade recursos de Pausar e Retroceder do player) . 4. Fazer anotações sobre o conteúdo. 5. Refazer os exemplos disponibilizados. 6. Anotar as dúvidas.	1. Solucionar as dúvidas que surgiram antes da aula. 2. Realizar a atividade de fixação. Disponível em: <http://bit.ly/2Gpx7Lf> 3. Fazer uma atividade prática com ferramenta. Atividade sugerida disponível em: <http://bit.ly/2S3cHcp>.	1. Acessar o objeto de aprendizagem listado a seguir para reforçar o conteúdo aprendido. Objeto disponível em: <http://bit.ly/2TOVW6r>. 2. Estudar o conteúdo da próximo aula.
Avaliação	Não se aplica	Atividade prática	Não se aplica

Fonte: Elaborada pelo autor.

Figura 26 – Planejamento da implementação de Flipped Classroom para Teste de Mutação

Nome do Tópico: Teste Baseado em Defeitos e o critério Teste de Mutação

Objetivos de aprendizagem para o tópico:
- Identificar a fonte de informação usada para condução da atividade;
- Conhecer o critérios teste de mutação;
- Compreender sobre o processo de aplicação do critérios Teste de Mutação na condução da atividade de teste;
- Aplicar o critério Teste de Mutação em programas implementados na linguagem C.

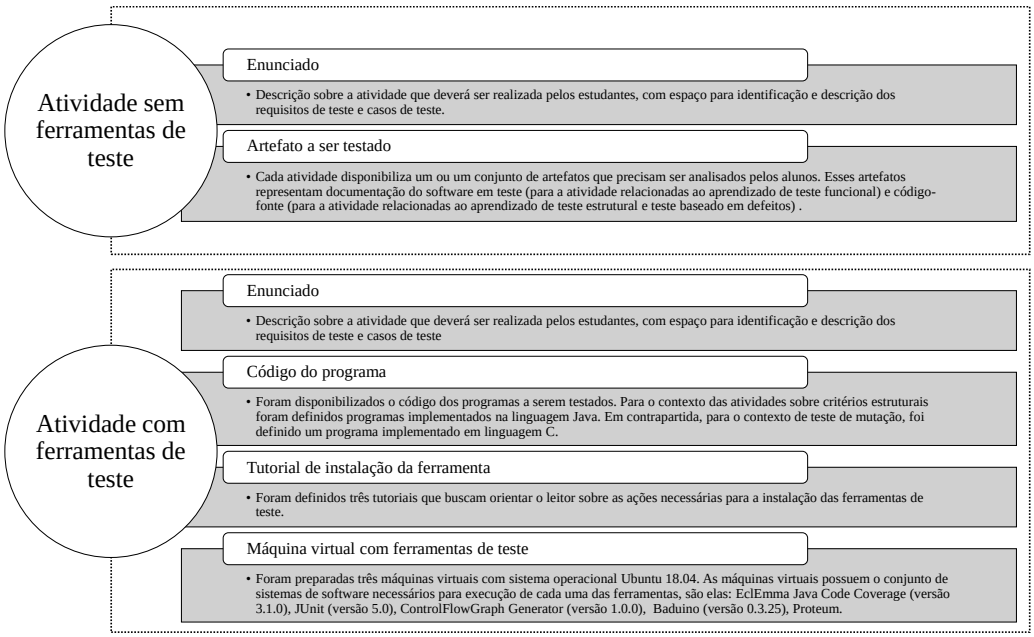
Recursos:
- Vídeos
  - Introdução ao Teste de Mutação. Disponível em: <http://bit.ly/2qmjTEX>
  - Teste de Mutação na Prática. Disponível em: <http://bit.ly/2yLrt0K>

	Antes da aula	**Durante a aula**	**Depois da aula**
Atividades	1. Acessar os mecanismos de apoio. 2. Eliminar as distrações (e.g. sair das redes sociais). 3. Assistir os vídeos (utilizar a vontade recursos de Pausar e Retroceder do player) . 4. Fazer anotações sobre o conteúdo. 5. Refazer os exemplos disponibilizados. 6. Anotar as dúvidas.	1. Solucionar as dúvidas que surgiram antes da aula. 2. Realizar a atividade de fixação. Disponível em: <http://bit.ly/2BA6qzF> 3. Fazer uma atividade prática com ferramenta. Atividade sugerida disponível em: <http://bit.ly/2Sy15mO>.	1. Estudar o conteúdo da próximo aula.
Avaliação	Não se aplica	Atividade prática	Não se aplica

Fonte: Elaborada pelo autor.

Durante a realização das atividades práticas de aplicação dos critérios, os alunos precisarão aprender a manusear algumas ferramentas de teste, bem como instalá-las. Para apoiar o aprendizado, alguns mecanismos de apoio foram produzidos pelo autor da pesquisa, os quais contemplam o processo de instalação e exemplos de utilização. Também foram preparadas máquinas virtuais para cada uma das ferramentas, com o propósito de minimizar o esforço necessário despendido pelo professor que deseja adotar o modelo *flipped classroom*. Visando ilustrar os constituintes das atividades, a Figura 27 foi desenvolvida.

Figura 27 – Constituintes das atividades que integram a abordagem FlipClass-STT

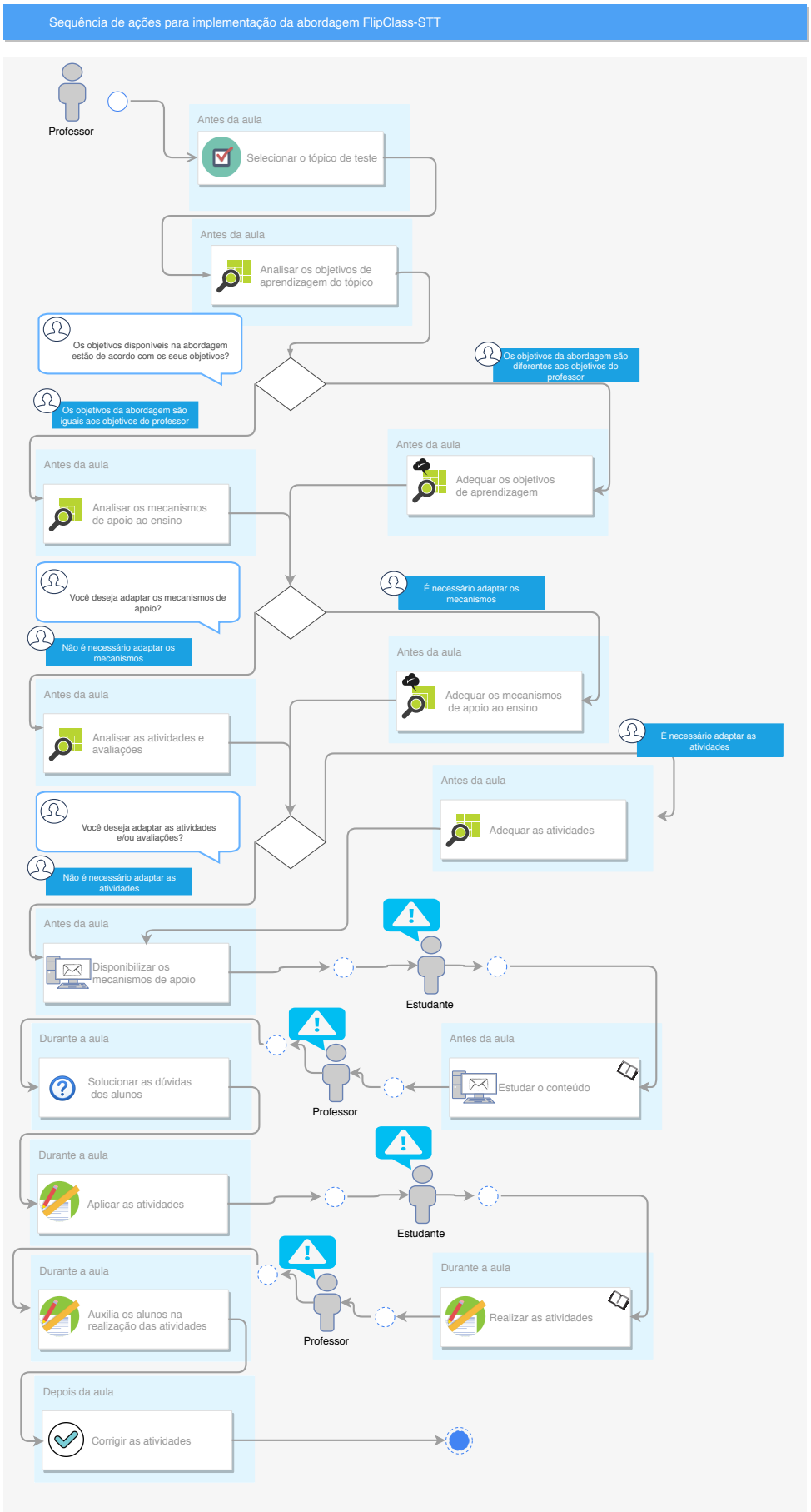


Fonte: Elaborada pelo autor.

A sequência de ações é constituída por diferentes caminhos que precisam ser ponderados pelo professor. Em especial, ele precisará analisar os objetivos de aprendizagem, o conteúdo da abordagem, as atividades e tomar decisões sobre a utilização e/ou adaptação dos mesmos. A sequência também indica o momento necessário de disponibilização do conteúdo aos alunos, em que será fundamental um ambiente virtual de aprendizagem ou sistema educacional similar para disponibilizar os recursos educacionais.

Outro aspecto importante que pode gerar esforço ao professor que deseja implantar *flipped classroom* refere-se à definição de uma estratégia para implementação (BERGMANN; SAMS, 2012), isto é, organizar a sequência de ações que devem ser feitas para a implementação efetiva do modelo. Assim, para complementar a abordagem, após a definição do planejamento de implementação da mesma, foi estabelecida uma sequência de ações (Figura 28) que devem ser realizadas pelos sujeitos envolvidos no processo de ensino e aprendizagem (professor e aluno). Em especial, essa sequência visa apoiar o professor durante a definição da implementação do modelo, seguindo a abordagem FlipClass-STT.

Figura 28 – Sequência de ações para implementação da abordagem. Versão maximizada disponível em: <http://bit.ly/2QzeY3g>



Fonte: Elaborada pelo autor.

4.4 Publicação e Aplicação da abordagem FlipClass-STT

A partir da definição da abordagem FlipClass-STT, um artigo científico foi redigido e publicado na Revista Novas Tecnologias na Educação (RENOTE) (PASCHOAL; SOUZA, 2018a). Na sequência, a abordagem foi aplicada em uma disciplina destinada ao ensino de teste de software. A aplicação foi conduzida com o propósito de observar a sua repercussão, sendo descrita em Paschoal e Souza (2018a). Durante essa aplicação, o *feedback* dos estudantes matriculados na disciplina foi coletado e com isso foi possível observar que a abordagem necessitava de um mecanismo para apoiar a sua implantação, especialmente relacionado ao conteúdo para disponibilização no ambiente virtual de aprendizagem. Nessa perspectiva, surgiu a necessidade de aprimorar a abordagem.

4.5 Aprimoramento da FlipClass-STT

Em uma aplicação preliminar da abordagem FlipClass-STT, relatada no artigo de Paschoal e Souza (2018a), foi possível observar que alguns alunos que estudaram com o material didático indicados na abordagem não ficaram totalmente satisfeitos com os recursos educacionais. Os alunos recomendaram que diferentes recursos educacionais sejam disponibilizados para o ensino, em múltiplos formatos. Alguns participantes do estudo também relataram que seria importante disponibilizar exercícios para os alunos realizarem "antes da aula", ao estilo de *quiz*. Nesse sentido, os alunos poderiam compreender qual o conhecimento esperado pelo professor durante o estudo.

Adicionalmente ao trabalho publicado em Paschoal e Souza (2018a), foi realizado um estudo com uma turma de alunos, visando entender a viabilidade do modelo *flipped classroom* a partir da abordagem FlipClass-STT. Ela foi utilizada por um professor da área de teste de software em uma disciplina específica sobre teste. No estudo que será discutido no próximo capítulo, foi possível observar que outros recursos educacionais poderiam ser contemplados na abordagem. Foi constatado que a abordagem ficou limitada aos recursos educacionais e às atividades pré-estabelecidas.

Ao mesmo tempo, há estudos que revelam que os recursos educacionais sobre teste de software disponíveis atualmente não vêm satisfazendo os professores da área (PASCHOAL; SOUZA, 2018b). Existem professores que têm dificuldade em definir atividades para serem utilizadas no ensino de teste (PASCHOAL; SOUZA, 2018b). Apesar disso, sabe-se que a comunidade de computação, em especial de teste de software, vem demonstrando interesse na produção de recursos educacionais aberto. Assim, acredita-se que essa comunidade pode contribuir com a definição de novos recursos educacionais e atividades, as quais podem também complementar e manter a abordagem sustentável.

Diante da problemática levantada, um mecanismo de apoio foi concebido, sendo

capaz de suportar o registro de professores que são autores de materiais didáticos e possibilitar que esses usuários façam o registro dos seus materiais didáticos. Esse sistema é um catálogo digital que busca facilitar a pesquisa por recursos educacionais abertos do domínio de teste de software. Ele não tem a intenção de funcionar como um repositório, mas busca oferecer suporte ao gerenciamento de recursos existentes. Desse modo, somente possibilita o cadastro das referências dos recursos educacionais, contendo o endereço da localização efetiva do conteúdo.

4.5.1 Catalog-STT: um Catálogo de Recursos Educacionais Abertos sobre Teste de Software

O sistema foi desenvolvido para complementar e aprimorar a abordagem FlipClas-STT, em que professores/autores de REA poderão contribuir com a disponibilização de novos recursos para a abordagem. Ao mesmo tempo, contribui de uma maneira mais ampla para a comunidade de professores de teste de software, os quais relatam dificuldade ao ensinar a aplicação prática da atividade. Adicionalmente, o sistema visa flexibilizar a abordagem, no sentido de tornar mais dinâmica a seleção do material, a coleta do *link* do recurso e a disponibilização do mesmo no ambiente virtual de aprendizagem.

Ele foi desenvolvido utilizando PHP (*Hypertext Preprocessor*), MySQL, JavaScript, CSS3 e HTML.Ressalta-se que durante o desenvolvimento do sistema, foi adotado o *framework* Materialize[7] para implementação dos módulos que o constituem. Essa seção descreve o sistema, suas principais funcionalidades e a estratégia adotada pelo autor desta pesquisa para acompanhar a evolução da abordagem.

Definição do sistema

O sistema desenvolvido é denominado Catalog-STT (*A Catalog of Open Educational Resources to Support Software Testing Teach*) e possui a habilidade de suportar múltiplos navegadores (*cross-browser*), seguindo as especificações do W3C (*World Wide Web Consortium*[8]). Ele é capaz de registrar recursos educacionais abertos sobre o domínio de teste de software.

Catalog-STT trata todos os procedimentos de registro de novos recursos educacionais – seguindo um padrão de metadado especializado, a busca por recursos educacionais que estão disponíveis no catálogo, a atualização do registro de recursos educacionais pré-existentes, a notificação ao usuário sobre a existência de atualização de recursos educacionais. O domínio do software ficou restrito ao processo de registro de recursos educacionais abertos que já estão previamente hospedados em repositórios. Nesse sentido, o sistema

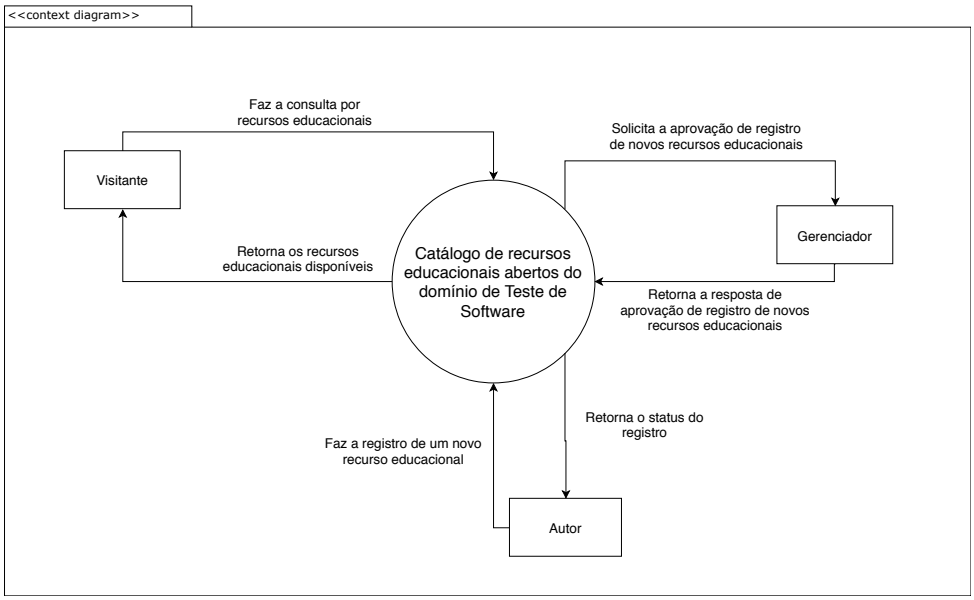
[7] Mais informações disponíveis em: <https://materializecss.com/>
[8] Mais informações disponíveis em: <https://www.w3.org/standards/>

serve como um mecanismo que centralizará os recursos educacionais disponíveis por autores de teste de software.

Catalog-STT também retorna resultados sobre os recursos educacionais registrados no sistema, os quais foram catalogados previamente seguindo um metadado para repositórios de objetos de aprendizagem adaptado do IEEE (*Institute of Electrical and Electronics Engineers*). Para tanto, o sistema recebe um conjunto de parâmetros de entrada, com o propósito de retornar um conjunto de recursos educacionais abertos cadastrados no sistema.

Para contextualizar o sistema, foi desenvolvido um diagrama de contexto (Figura 29), que mostra a interface do software Catalog-STT com os diferentes tipos de usuários que interagem com o sistema (*i.e.*, visitante, autor e gerenciador). No diagrama, é possível visualizar que há comunicação entre os dados dos usuários, isto é, alguns dados são fornecidos por um tipo de usuário para o sistema e, consequentemente, aqueles dados são oferecidos pelo sistema a outro tipo de usuário. Em especial, o diagrama descreve os dados que o sistema absorve e deve processar e os dados que o sistema oferece aos utilizadores.

Figura 29 – Modelagem do escopo do sistema Catalog-STT



Fonte: Elaborada pelo autor.

Classes de usuários

Catalog-STT possui três tipos de usuários: visitante, autor, gerenciador. A seguir são apresentadas as principais características do grupo de usuários que é esperado pelo sistema.

Visitante: entende-se por visitante o usuário que não possui um registro no sistema. Ele pode realizar consultadas na base de dados do sistema, com o propósito de buscar os recursos educacionais abertos que estão catalogados (*i.e.*, submetidos pelos autores e aprovados pelo gerenciador).

Autor: entende-se por autor o usuário que se cadastra no sistema e realiza o registro ou gerenciamento de seus próprios recursos educacionais. O autor pode registrar a quantidade que desejar de recursos educacionais. Ele que será responsável por manter os metadados dos seus recursos educacionais atualizados. Em caso de atualização do recurso educacional, ele que deverá registrar a atualização.

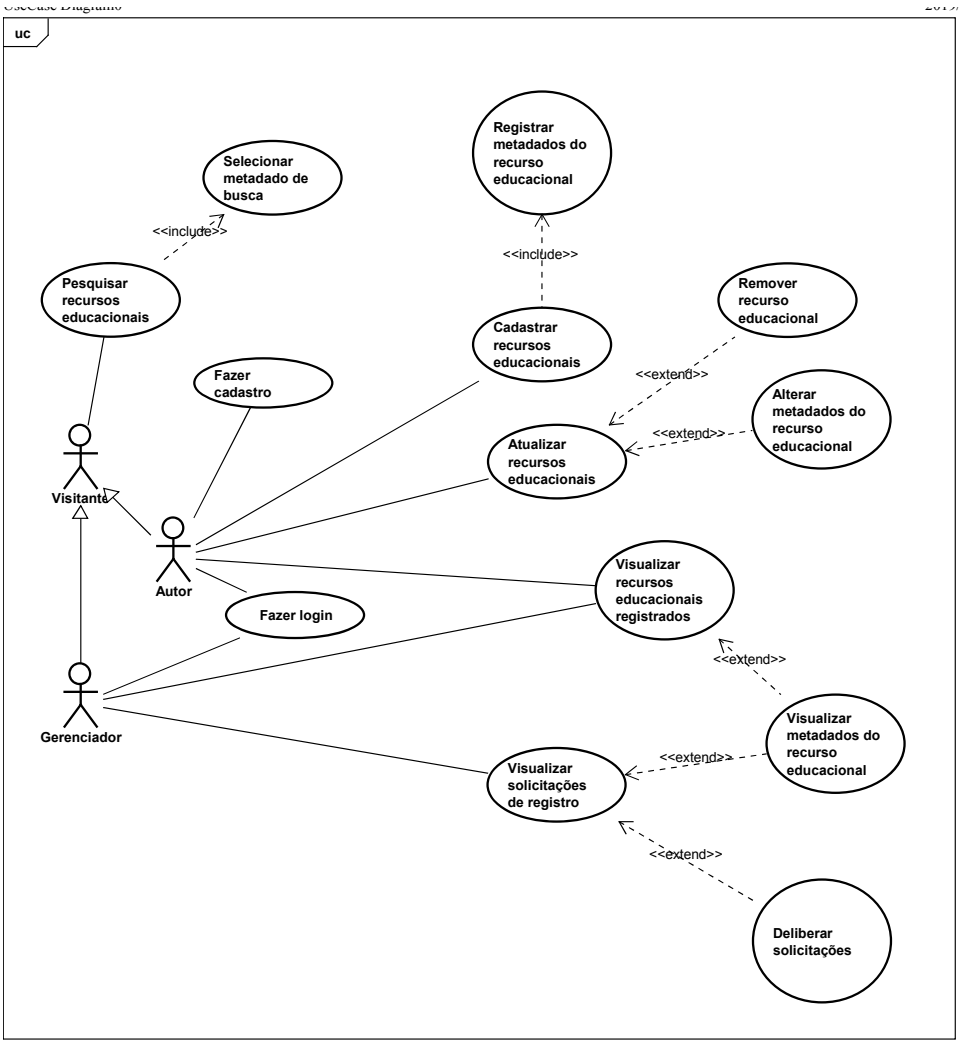
Gerenciador: o gerenciador é o administrador do sistema. O gerenciador faz o gerenciamento dos recursos educacionais registrados pelos autores. Ele que analisa o registro dos recursos educacionais e altera o status do recurso para aprovado ou reprovado. Esse usuário será o autor desta pesquisa, uma vez que foi definido como estratégia para evitar a aprovação de REA que não atendem os objetivos de aprendizagem estabelecidos no decorrer da Seção 4.2.2.

Módulos

O objetivo principal de Catalog-STT é reunir os recursos educacionais abertos desenvolvidos por professores da área de teste de software, na condição de poderem ser utilizados na abordagem FlipClass-STT. A fim de atender essa finalidade, o professor/autor do REA deve fornecer ao sistema dados que caracterizam o REA. Nessa perspectiva, Catalog-STT opera coletando esses dados, armazenando-os previamente em um banco de dados. A catalogação dos REA é condicionada a aprovação do autor desta pesquisa, que no sistema possui o perfil de gerenciador. Após a aprovação do administrador do sistema, os REA catalogados são apresentados ao usuário na forma de lista ordenada de REA, onde é possível visualizar, realizar buscas e expandir os resultados para cada REA listado.

As funcionalidades do sistema são baseadas em interação. Essa interação entre o sistema e usuários foi representada por um diagrama de casos de uso, construído com notação UML (*Unified Modeling Language*). Na Figura 30 são narradas as interações entre o sistema e os atores envolvidos. Uma descrição completa sobre os casos de uso do sistema está disponível em: <http://bit.ly/2N6fm4l>, em conjunto com os requisitos do sistema. Diante disso, as funcionalidades do sistema foram especializadas em três módulos, a saber: registro, seleção e gerenciamento. Cada um desses módulos é descrito em detalhes a seguir:

Figura 30 – Diagrama de Casos de Uso do sistema Catalog-STT



Fonte: Elaborada pelo autor.

Módulo de Registro

O módulo de registro remete a duas opções de funcionamento. A primeira opção, está relacionada com o cadastro de autores, em que é necessário o preenchimento de um formulário que contém campos para disponibilização de dados para acesso ao sistema (*e.g.*, e-mail, nome, senha, dentre outros). A segunda opção está associada com a inserção de REA pelos autores. Somente usuários registrados como autores tem acesso a segunda funcionalidade. Essa funcionalidade remete à entrada de dados por meio de um formulário, que contém os metadados do recurso educacional. Contudo, o registro deste novo REA aparecerá como pendente para o autor, permanecendo no banco de dados temporário, que necessita da aprovação do gerenciador do sistema para realmente se consolidar no referatório. Se o autor desejar fazer o cadastro de um novo REA, um outro formulário é exibido para obtenção dos metadados do novo REA. Esse novo REA ficara inserido em um banco de dados temporário, esperando a aprovação do gerenciador.

Módulo de Seleção

O módulo de seleção está relacionado com a entrada de informações sobre o conteúdo do REA que o usuário da abordagem FlipClass-STT deseja utilizar em uma aplicação da mesma. A entrada dessas informações é feita por meio de um formulário e filtros de busca. Nesse formulário, o usuário poderá inserir características que ele deseja que REA contenha. Essas características são mapeadas nos atributos dos metadados do REA. Assim, os resultados da busca estão relacionados com as informações de entrada do módulo de seleção, pois estas informações serão utilizadas no algoritmo de seleção para serem cruzadas com os metadados dos REA registrados no sistema.

Gerenciamento

Esse módulo permite que o autor desta pesquisa, desenvolvedor da abordagem FlipClass-STT, administre os REA que são previamente catalogados no sistema. Nessa perspectiva envolve o gerenciamento do conteúdo, em que é fornecida a opção para controlar a inserção de novos recursos, por meio da remoção de REA que não apresentam o conteúdo adequado ao catálogo e a abordagem FlipClass-STT. Além disso, permite a edição dos metadados, atualização desse metadados e emissão de avisos aos autores que possuem REA cadastrados sobre a necessidade de atualização dos metadados dos REA.

Metadados

Visando oferecer suporte à gestão de conteúdo educacional e facilitar a pesquisa por recursos educacionais abertos, o sistema de catalogação utiliza um padrão de metadados que auxilia na estruturação e categorização do conteúdo. De acordo com Jayal (2010), metadados contêm uma descrição precisa sobre os dados e podem ser usados para localizar e gerenciar os dados. Assim, um metadado descreve as características do REA que são utilizadas para sua catalogação.

De acordo com Dutra, Tarouco e Passerino (2011), os recursos educacionais são melhor aproveitados quando estão organizados em uma classificação de metadados. Esses padrões contribuem com a catalogação em repositórios de objetos de aprendizagem e auxiliam a filtragem de uma busca, facilitando a sua recuperação (FERLIN *et al.*, 2010). Com base nisso, através de um sistema de consulta, o usuário poderá localizar o conteúdo adequado para utilização.

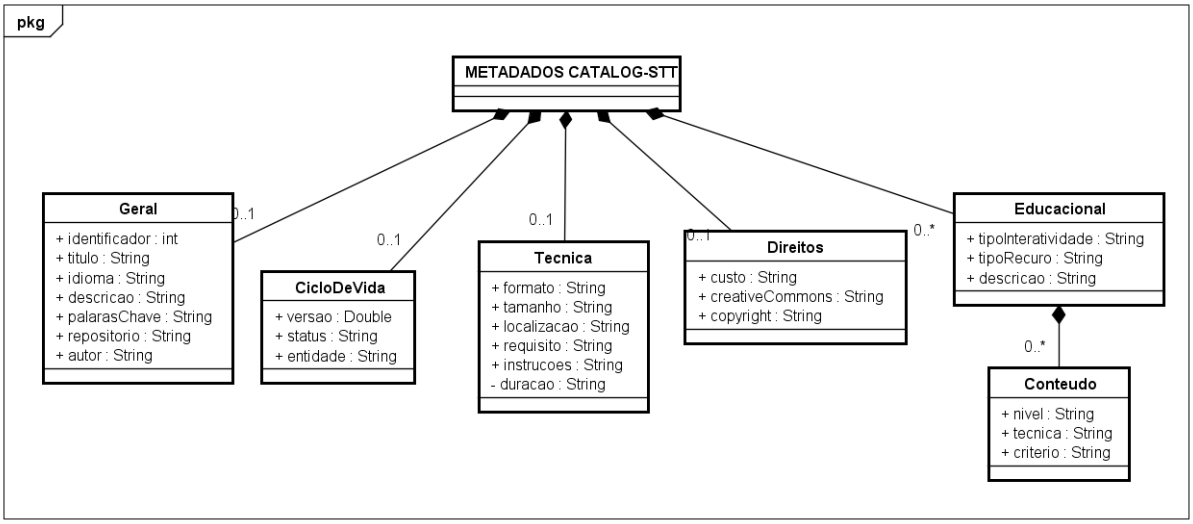
Atualmente, existem diversos padrões de metadados. Em especial, no contexto educacional, organizações se especializaram em criar especificações que orientam a estrutura para metadados de objetos de aprendizagem, os quais podem facilmente ser utilizados para caracterizar os recursos educacionais abertos. Dentre as especificações existentes, destacam-se os padrões IEEE-LOM e Dublin Core (VICARI *et al.*, 2010). Outras especificações para descrição de objetos de aprendizagem existentes baseiam-se na especificação

do IEEE-LOM, descrita no documento IEEE 1484.12.1-2002[9]. O padrão de metadados IEEE LOM é descrito por Vicari *et al.* (2010) como um modelo completo, que estrutura o conjunto de metadados em nove categorias. Além disso, ele é específico para caracterização de objetos de aprendizagem, diferentemente do padrão Dublin Core, que é destinado para categorizar qualquer tipo de recurso digital. Assim, devido à sua ampla aceitação e adoção, o padrão IEEE-LOM foi elegido como base principal para a estrutura de metadados do sistema Catalog-STT.

O padrão IEEE-LOM é definido pelo grupo de trabalho da IEEE como flexível e pode ser adaptado, em conformidade com o repositório ou referatório que irá utilizá-lo. Nesse sentido, é necessário analisar as especificações descritas no documento IEEE 1484.12.1-2002. Durante o desenvolvimento do repositório CESTA (Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem)[10], por exemplo, foi necessário adaptar o padrão LOM, em conformidade com o que a equipe de professores acreditavam ser necessárias, resultando na eliminação de alguns atributos e seleção de outros.

No contexto do desenvolvimento de Catalog-STT foi necessário estudar todos atributos das nove categorias. Na sequência, foi definido uma versão do modelo IEEE-LOM que foi utilizada no desenvolvimento do sistema. A Figura 31 ilustra as categorias em que os metadados foram agrupados. Nas Tabelas 31 e 32, os elementos que constituem o padrão adotado são apresentados e descritos.

Figura 31 – Estrutura de metadados adotada



Fonte: Elaborada pelo autor.

[9] Mais informações disponíveis em: <https://ieeexplore.ieee.org/document/1032843>
[10] Mais informações disponíveis em: < http://www.cinted.ufrgs.br/CESTA/>

Tabela 31 – Descrição dos metadados

Identificador	Nome	Descrição
1	Geral	A categoria é constituída por um conjunto de informações gerais que descrevem o recurso educacional como um todo
1.1	Identificador	Uma identificação exclusiva do recuso educacional
1.2	Título	Nome dado ao recurso educacional
1.3	Idioma	Idioma do recurso educacional
1.4	Descrição	Uma descrição textual sobre o conteúdo disponível no recurso educacional
1.5	Palavras-chave	Um conjunto de palavras que caracterizam o recurso educacional
1.6	Repositório	O nome do repositório em que o recurso educacional está hospedado
1.7	Autor	O nome dos autores do recurso educacional. Entende-se por autores aqueles que contribuíram com a sua concepção, desenvolvimento e/ou avaliação
2	Ciclo de Vida	A categoria é constituída por um conjunto de informações sobre o histórico e estado atual do recurso educacional
2.1	Versão	A edição do estado atual do recurso educacional
2.2	Status	O estado atual do recurso educacional (disponível, indisponível)
2.3	Entidade que contribuiu	Uma descrição sobre as entidades que contribuíram com o desenvolvimento do recurso educacional (*e.g.*, grupo de pesquisa, instituição de ensino)
3	Técnica	A categoria é constituída por um conjunto de informações técnicas sobre os requisitos e característica do recurso educacional
3.1	Formato	Formato do tipo de arquivo do recurso educacional (*e.g.*, .MP4, .EXE). Esse elemento é necessário para apoiar a identificação do tipo de software necessário para execução do recurso educacional
3.2	Tamanho	Tamanho do recurso educacional
3.3	Localização	O endereço onde o recurso educacional está disponível. Em geral é dado por uma URL (*Universal Resource Locator*)
3.4	Requisitos tecnológicos	Esse elemento contém os requisitos tecnológicos para utilização do recurso educacional, tais como a indicação do software necessário para execução e/ou manipulação, versão mínima do software para execução, etc
3.5	Instruções de instalação	Descrição sobre o processo de instalação do recurso educacional

Fonte: Elaborada pelo autor.

Tabela 32 – Descrição dos metadados (Continuação)

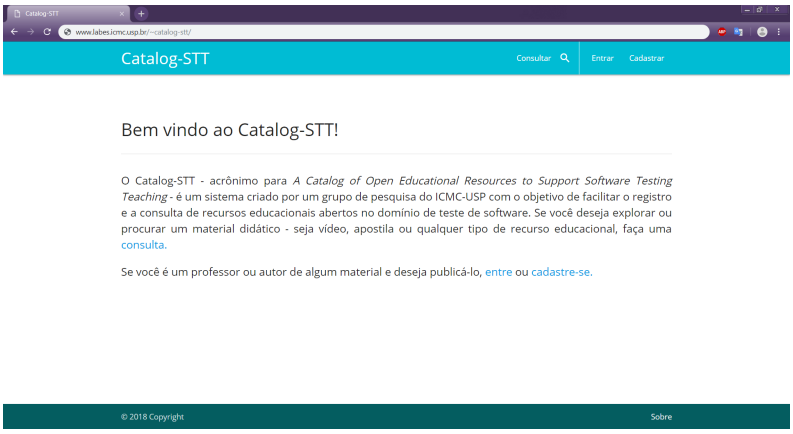
3.6	Duração	Esse elemento deve ser utilizado apenas para recursos educacionais que possuem tempo de duração, como vídeo e sons. O elemento contém informações relacionadas ao tempo total de duração do vídeo ou do áudio
4	Educacional	A categoria é constituída por um conjunto de informações sobre características educacionais e pedagógicas do recurso educacional
4.1	Tipo de interatividade	Elemento que indica o tipo de interação que recurso educacional oferece (expositiva, participativa/ativa, mista)
4.2	Tipo de recurso de aprendizagem	Tipo especifico do recurso educacional (*e.g.*, exercício, simulação, infográfico, slide, vídeo, teste de conhecimento, textos, jogos, módulo educacional)
4.3	Conteúdo	Essa subcategoria é constituída por um conjunto de informações específicas sobre o conteúdo do recurso educacional
4.3.1	Nível de teste	Esse elemento contém uma descrição sobre os níveis de teste que são discutidos no recurso educacional (*e.g.*, teste de unidade, teste de integração, teste de sistema, teste de aceitação)
4.3.2	Técnica de teste	Esse elemento contém uma descrição sobre as técnicas de teste que são discutidas no recurso educacional (*e.g.*, teste funcional, teste estrutural, teste baseado em defeitos, teste baseado em modelos)
4.3.3	Critério de teste	Esse elemento contém uma descrição sobre os critérios de teste que são discutidos no recurso educacional (*e.g.*, particionamento em classes de equivalência, grafo causa-efeito, todos-usos, potenciais usos, todos-nós, teste de mutação, dentre outros)
4.4	Descrição	Indicações dos autores para utilização pedagógica do recurso educacional
5	Direitos	Essa categoria contém informações sobre direitos de propriedade intelectual e condições de uso do recurso educacional
5.1	Custo	Informação referente a necessidade de custo para uso do recurso educacional, isto é, se a utilização do recurso educacional exige algum tipo de pagamento.
5.2	Creative Commons	Indicação do tipo de licença de uso do recurso educacional
5.3	Copyright	Indicação de copyright (se aplicável)

Fonte: Elaborada pelo autor.

Funcionalidades

Nesta seção, as principais funcionalidades oferecidas pelo sistema Catalog-STT são descritas em detalhes. A página inicial do sistema é apresenta na Figura 32. Ela contém uma contextualização sobre o sistema e suas funcionalidades, um conjunto de opções com *links* para acessar as funcionalidades da mesma, como a opção de consulta destinada a busca de REA catalogados. Também inclui a opção para o usuário fazer *login* no sistema e o cadastro de usuários.

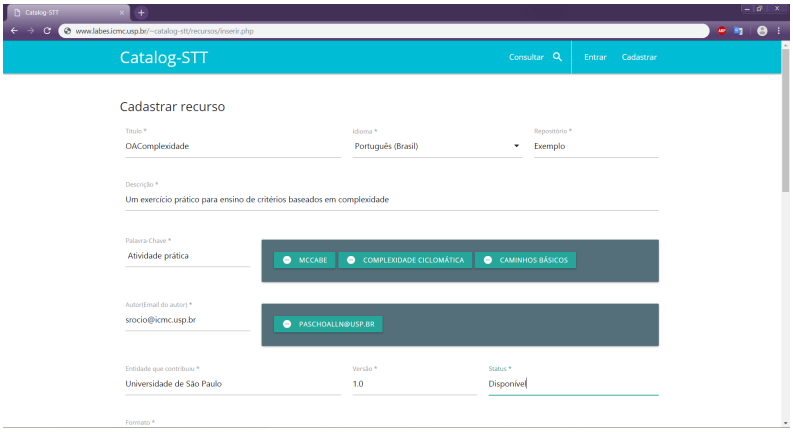
Figura 32 – Interface da página inicial do sistema Catalog-STT



Fonte: Elaborada pelo autor.

Uma das principais funções é o registro de recursos educacionais abertos. A Figura 33 apresenta uma captura de tela do sistema, em que são oferecidos alguns dados de exemplo. Nota-se que os atributos do metadado apresentado na Figura 31 compõem o formulário de registro. Durante o registro, o sistema exige ao autor que todos os campos sejam preenchidos.

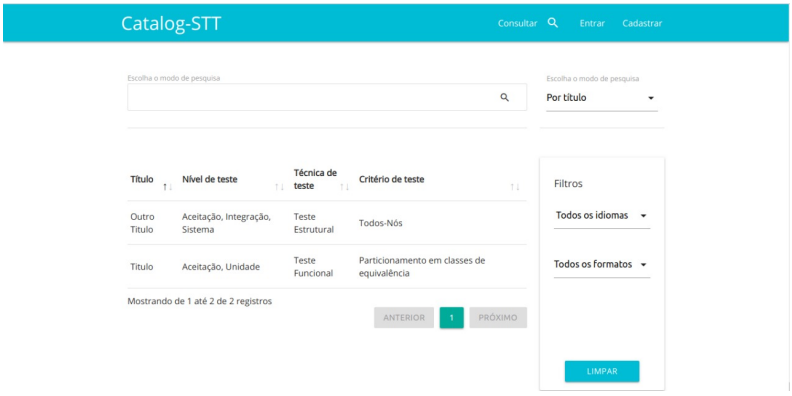
Figura 33 – Interface de registro de REA do sistema Catalog-STT



Fonte: Elaborada pelo autor.

Outra funcionalidade principal do Catalog-STT é a recuperação de recursos educacionais que estão registrados pelos autores. A Figura 34 apresenta a interface que foi capturada do sistema. Nessa tela, os recursos cadastrados no sistema serão listados. O usuário poderá classificar a apresentação, por titulo do REA e conteúdo (nível de teste, técnica e critério de teste). Há algumas opções para filtro, em que o usuário pode procurar REA em linguagens específicas e formato de recurso. Além disso, o usuário pode utilizar um campo de busca para procurar REA, podendo fazer buscas por título, autor e palavras-chave.

Figura 34 – Interface de consulta de REA do sistema Catalog-STT



Fonte: Elaborada pelo autor.

4.6 Considerações Finais

Este capítulo apresentou uma abordagem para definição e uso de *flipped classroom* no ensino de teste de software. A necessidade da abordagem surgiu a partir de observações feitas em estudos da área, os quais relatam que teste de software é normalmente ensinado com o método tradicional de ensino e este não está conseguindo satisfazer os estudantes atuais. Assim, a abordagem FlipClass-STT foi desenvolvida, seguindo os direcionamentos de Bergmann e Sams (2012). Ela é constituída por uma planejamento que contém os objetivos de aprendizagem, um conjunto de mecanismos de apoio, um conjunto de atividade, um artefato que representa as ações que são realizadas pelos sujeitos envolvidos na implementação da abordagem e um sistema que busca apoiar o professor a selecionar os recursos educacionais e ao mesmo tempo contribui com a sustentabilidade da abordagem.

Após a definição da abordagem, ela foi utilizada em uma disciplina destinada ao ensino de teste de software. A aplicação da mesma foi realizada visando compreender melhor o processo de aplicação e de sua viabilidade para implantação do modelo *flipped classroom*. O próximo capítulo é destinado a apresentação do estudo sobre uma experiência de implementação da abordagem.

CAPÍTULO

5

APLICAÇÃO DA ABORDAGEM FLIPCLASS-STT

5.1 Considerações Iniciais

A viabilidade sobre o ensino de teste de software com o modelo *flipped classroom* foi uma temática abordada nesta pesquisa. Por razão da área de teste de software ser ampla e existirem diversas variáveis associadas ao processo de ensino e aprendizagem, a pesquisa apresentada no Capítulo 3 limitou-se a investigar o modelo utilizado no ensino de teste funcional e particionamento em classes de equivalência. Na sequência das observações sobre a viabilidade do modelo, uma abordagem para apoiar a implementação do mesmo no ensino de alguns tópicos de teste de software foi delimitada. Esses tópicos além de compreender ensino de teste funcional e particionamento em classes de equivalência abrangem conteúdos não investigados durante a análise de viabilidade.

A abordagem FlipClass-STT, por sua vez, foi definida para apoiar a implementação do modelo em cursos e disciplinas de engenharia de software e teste de software, podendo ser utilizada integralmente ou parcialmente pelo professor interessado nela. Assim, visando contribuir com o ensino de teste de software realizado no Instituto de Ciências Matemáticas e de Computação, a abordagem foi aplicada no curso de graduação em Sistemas de Informação. A proposta inicial foi utilizar a abordagem para compreender melhor o fluxo de trabalho necessário para sua implementação e reconhecer as suas limitações. Com base nisso, melhorias na abordagem foram identificadas e aperfeiçoamentos foram realizados, conforme descrito na Seção 4.5.

Ao mesmo tempo que abordagem é aplicada, é possível ter uma visão sobre o impacto do modelo pedagógico *flipped classroom* nos outros tópicos de teste de software que são contemplados pela abordagem. No entanto, a visão sobre o impacto está condicionada a implantação da abordagem, não podendo ser generalizada para o modelo. Assim,

ao passo que a abordagem foi aplicada na turma, é possível observar, por exemplo, quanto o aluno aprende durante o processo de ensino. Com base nisso, estima-se conseguir obter evidências sobre os benefícios da abordagem para a aprendizagem dos estudantes.

Este capítulo é dedicado ao relato de uma experiência conduzida, em que a abordagem FlipClass-STT é utilizada para implementação do modelo *flipped classroom*. Para apresentá-lo, ele foi estruturado da seguinte forma. A Seção 5.2 retrata o contexto do relato de experiência, em que a disciplina, os conteúdos planejados para serem ensinados e as atividades previstas e realizadas são apresentados. Na Seção 5.3 é descrito o planejamento detalhado sobre como foram especificados os instrumentos utilizados para coletar os dados dos participantes. Na sequência (Seção 5.4) é relatada a experiência. O impacto do uso da abordagem no aprendizado dos alunos é discutido na Seção 5.5. Durante a Seção 5.6 são abordadas as lições aprendidas. O desfecho de todos os materiais utilizados no relato de experiência e dados brutos coletados é discutido na Seção 5.7. Em seguimento, na Seção 5.8, os dados do relato da experiência são confrontados com dados históricos. Por fim, as considerações finais são retratadas.

5.2 Proposta Didática

Buscando compreender melhor os aspectos associados a utilização da abordagem FlipClass-STT, optou-se por realizar uma aplicação da mesma em uma disciplina específica, em que todo o conteúdo contemplado pela abordagem pudesse ser ensinado. Entende-se que a aplicação por si só não configura uma avaliação da abordagem, mas é capaz de exteriorizar alguns prós sob a mesma e melhorias a serem realizadas.

A abordagem foi aplicada no 2º semestre de 2018 na disciplina de Tópicos Especiais em engenharia de software, oferecida aos alunos do 4º ano do curso de Bacharelado em Sistemas de Informação do ICMC/USP. A carga horária é de 30 horas, divididas em 2 aulas semanais. O tópico escolhido para a disciplina foi teste de software e, portanto, foi possível aplicar a abordagem FlipClass-STT.

Durante o segundo semestre de 2018, em particular, a disciplina foi oferecida com duração de 15 semanas, no período noturno, todas as sextas-feiras, das 21 horas até as 22 horas e 40 minutos. Com base nas características da disciplina, um cronograma que contém o planejamento da disciplina foi redigido e a proposta pedagógica foi delimitada.

Conforme Fassbinder *et al.* (2015) não é recomendado usar *flipped classroom* em toda a disciplina. Os alunos estão acostumados a estudar através do modelo tradicional e podem ficar intimidados ao se deparar com a nova maneira como o conteúdo é abordado. Além disso, eles podem ter dificuldade em acompanhar o ritmo de aulas invertidas. Por conta disso, durante o planejamento da disciplina foi definido que a disciplina utilizaria o modelo *flipped classroom* intercalando aulas com o modelo tradicional. Assim, o

planejamento da disciplina seguiu o roteiro apresentando na Tabela 33.

Tabela 33 – Resumo do desenvolvimento da disciplina

Sequência	Tópico	Descrição	Modelo pedagógico
Aula 1	Apresentação da disciplina	Aula destinada à apresentação da disciplina e discussão sobre o conteúdo que será abordado (ementa da disciplina), metodologia utilizada no ensino (modelo pedagógico), avaliações previstas, formas de avaliação e cronograma da disciplina.	Tradicional
Aula 2	Qualidade de software	Aula especifica sobre conceitos relacionados a qualidade de software e atividades de verificação e validação (atividades de análise estática e de analise dinâmica).	
Aula 3	Conceitos e terminologia de teste	Sessão designada para fazer preleções sobre definição de teste de software, apresentação de termos de jargão e descrição sobre a condução da atividade de teste.	
Aula 4	Teste estrutural: critérios baseados em fluxo de controle	Sessão dedicada a resolução de dúvidas dos alunos sobre os critérios baseados em fluxo de controle e condução de atividade prática em equipe, com supervisão do professor, para estimular o aprendizado de critérios Todos-Nós e Todas-Arestas.	*Flipped Classroom*
Aula 5		Aula dedicada a realização de atividades práticas em laboratório com as ferramentas JUnit, ControlFlowGraph Generator, EclEmma Java Code Coverage.	
Aula 6		Sessão dedicada a resolução de dúvidas, revisão do conteúdo e atividades avaliativas.	Tradicional
Aula 7	Teste estrutural: critérios baseados em fluxo de dados	Sessão dedicada a resolução de dúvidas dos alunos sobre os critérios baseados em fluxo de dados e condução de atividade prática em equipe, com supervisão do professor, para estimular o aprendizado de critérios Todas-Definições, Todos-Usos, TodosPotenciais-Usos.	*Flipped Classroom*
Aula 8		Aula dedicada a realização de atividades práticas em laboratório com a ferramenta Baduino.	
Aula 9		Sessão dedicada a resolução de dúvidas, revisão do conteúdo e atividades avaliativas.	Tradicional
Aula 10	Teste baseado em defeitos: análise de mutantes	Sessão dedicada a resolução de dúvidas dos alunos sobre testes baseados em defeitos e condução de atividade prática em equipe, com supervisão do professor, para estimular o aprendizado de teste de mutação.	*Flipped Classroom*
Aula 11		Aula dedicada a realização de atividades práticas em laboratório com a ferramenta Proteum.	
Aula 12		Sessão dedicada a resolução de dúvidas, revisão do conteúdo e atividades avaliativas.	Tradicional
Aula 13	Teste funcional: particionamento em classes de equivalência e análise do valor limite	Sessão dedicada a resolução de dúvidas dos alunos sobre teste funcional, condução de atividade prática em equipe, com supervisão do professor, para estimular o aprendizado de particionamento em classes de equivalência e análise do valor limite.	*Flipped Classroom*
Aula 14		Resolução de atividade de aplicação de critérios de teste, resolução de dúvidas e revisão do conteúdo.	Parcialmente *Flipped Classroom*
Aula 15	Avaliação final	Aula dedicada a avaliação final da disciplina por meio de uma prova.	Tradicional

Fonte: Elaborada pelo autor.

Durante o planejamento da disciplina, foram pré-definidas estratégias para acompanhar o aprendizado dos alunos. Assim, foi determinado que o aprendizado seria mensurado por meio de testes de conhecimento, similares aos utilizados no segundo experimento, mencionado na Seção 3.3. Para o contexto da avaliação na disciplina, além das notas nos testes de conhecimento, foram utilizadas as notas obtidas pelos alunos no decorrer de atividades de participação e dos trabalhos práticos associados a aplicação de critérios e uso de ferramentas de automatização. No planejamento, foi previsto que os alunos que não atingiram a nota mínima (*i.e.* 5 pontos) nessas avaliações, deveriam fazer uma prova final.

5.3 Coleta dos Dados

Com o intuito de compreender melhor a repercussão do aprendizado de teste de software frente ao uso de *flipped classroom* com a abordagem FlipClass-STT, algumas estratégias foram adotadas. Essas estratégias buscam acompanhar o aprendizado adquirido pelo aluno no decorrer da experiência. O aprendizado é a variável observada no relato de experiência, uma vez que reflete o resultado do que está sendo ensinado. Em função

disso, se o aluno consegue aprender por meio da iniciativa estabelecida, a maneira como o conhecimento está sendo transmitido ao aluno é considerada preliminarmente satisfatória.

De acordo com Ausubel (1980) *apud* Medina (2004), o fator isolado mais importante para o aprendizado é o conhecimento preliminar do aluno. Por consequência, o que o aluno já sabe é uma particularidade que deve ser considerada durante o planejamento didático. Nessa perspectiva, no início do semestre foi planejado um questionário com o propósito de coletar informações sobre o conhecimento prévio do estudante em relação ao tema teste de software.

O questionário foi elaborado com base no questionário escrito por Melo (2018) e é composto por por cinco questões objetivas e uma questão dissertativa (Tabela 34). Dentre as opções de respostas das questões objetivas, foram atribuídas as seguintes alternativas: 1 - Nenhuma experiência; 2 - Estudei em casa ou livro; 3 - Pratiquei em 1 projeto em sala de aula; 4 - Usei em 1 projeto na indústria; 5 - Usei em vários projetos na indústria.

Tabela 34 – Questões que compõem o formulário de caracterização de *expertise* dos estudantes

Questão	**Tipo**
1. Como você avalia seu grau de conhecimento em Engenharia de Software?	Objetiva (Obrigatória)
2. Como você avalia seu grau de conhecimento em Teste de Software?	Objetiva (Obrigatória)
3. Você já teve a experiência de aplicar os critérios de Teste Funcional em projetos de software?	Objetiva (Obrigatória)
4. Você já teve a experiência de aplicar os critérios de Teste Estrutural em projetos de software?	Objetiva (Obrigatória)
5. Você já teve a experiência de aplicar o critério Teste de Mutação em projetos de software?	Objetiva (Obrigatória)
6. Se você já trabalhou com teste de software, descreva sobre as atividades que você desenvolvia: - Nome da empresa que trabalhou com teste de software (especificar se é contratado ou estagiário). - Quanto tempo? - Quais sistemas de software utilizou para apoiar a condução da atividade? - Quais os tipos de sistema você testou? Quais os domínios da aplicação (aplicação web, mobile, concorrente, etc.)? - Qual a fonte de informação você usou para testar o software (documentação ou código-fonte)? - Você usou algum critério e/ou alguma técnica de teste?	Descritiva (Não obrigatória)

Fonte: Elaborada pelo autor.

Tendo em vista que a abordagem FlipClass-STT é constituída por recursos educacionais abertos em formatos específicos (*i.e.* vídeos, objetos de aprendizagem), durante o planejamento das aulas foi elaborado um formulário para que os alunos pudessem classificar quais os recursos educacionais que eles possuíam preferência. Foram delimitados 12 diferentes tipos de recursos educacionais, a saber: 0 – Apresentação de slides; 1 – Artigos; 2 – Áudios (*podcasts*; 3 – Capítulos de livro; 4 – Infográficos; 5 – Jogos; 6 – Livros; 7 – Mapa conceitual/ Esquemas; 8 – Páginas web; 9 – Simulações; 10 – Tutoriais; 11 – Vídeos.

O formulário elaborado também contou com outras duas questões. Em uma delas, os alunos deveriam indicar a preferência em estudar com material didático impresso. Por outro lado, na outra alegação, os alunos deveriam apontar o quanto eles optavam por estudar com material didático digital. As assertivas foram definidas com base na escala Likert de seis pontos, que variam entre discordo totalmente até concordo totalmente. É importante relatar que esse formulário foi definido para caracterizar somente as preferências da turma de alunos por diferentes formatos. Em um primeiro momento, não foi ponderado a recomendação alternativa de diferentes formatos considerando as preferências individuais dos estudantes.

Em decorrência de se optar por fazer o acompanhamento do aprendizado dos alunos, foi necessário estabelecer instrumentos para realizar a supervisão. Assim, tendo como base o segundo experimento (apresentado em na Seção 3.3), foi definido que os alunos seriam analisados com base nos números de acerto em testes de conhecimento. Assim, foi necessário planejar testes de conhecimento (pré-teste, teste intermediário e pós-teste) para cada tópico contemplado pela abordagem. Esses testes de conhecimento são constituídos por um conjunto de questões objetivas, adaptadas especialmente de edições de exames de certificação da área de teste de software, como o CTFL (*Certified Tester Foundation Level*).

Cada teste é composto por um conjunto de questões que contemplam aspectos teóricos e práticos sobre a técnica e os critérios de teste. De maneira similar ao segundo experimento, foi definido inicialmente um pré-teste para cada tópico e variações desses testes foram desenvolvidas.

Além de investigar a frequência de acertos nos testes de conhecimento, as opiniões dos alunos frente ao uso do modelo *flipped classroom* foram planejadas para serem coletadas. Para tanto, as questões abertas definidas no estudo de He *et al.* (2016) foram utilizadas, a saber: (i) Você gostou ou não do modelo *flipped classroom*? Por que? ; (ii) Qual é a sua principal reclamação sobre esta experiência e quais são suassugestões para melhorias?. O propósito da utilização dessas questões foi compreender se os alunos estavam gostando do ensino de teste de software com o modelo *flipped classroom* a medida que cada conteúdo foi ensinado.

Após o planejamento das aulas e dos materiais necessário para coletar os dados dos participantes, a disciplina foi ministrada. A próxima seção é dedicada a descrição sobre a experiência, relatado de modo geral como foi ocorrida, atividades realizadas, momentos em que os testes foram aplicados. Salienta-se que não foram coletadas informações sobre o momento "antes da aula" (*e.g.*, tempo que o aluno utilizou para estudar, quantas vezes ele assistiu os vídeos, se o aluno precisou de material complementar) e por isso, elas não serão descritas no decorrer da seção.

5.4 Relato de Experiência

Durante o segundo semestre de 2018, a disciplina de Tópicos Especiais em Engenharia de Software foi ministrada, em conformidade como planejamento descrito na Seção 5.2. Durante a primeira semana, realizou-se a apresentação da disciplina, em que foram discutidos os conteúdos que seriam ensinados ao longo do semestre, os modelos pedagógicos adotados e as avaliações previstas. No decorrer da aula, o formulário para caracterização dos participantes foi aplicado. Antes do fim da aula, uma atividade foi realizada, em que os alunos deveriam tentar descrever alguns conceitos sobre teste de software. Essa atividade buscou estimular a participação dos estudantes e iniciar uma demonstração sobre conceitos fundamentais para a área de teste de software, os quais os participantes deveriam aprender no decorrer da disciplina.

A segunda aula foi dedicada ao ensino de atividades de verificação e validação e suas contribuições para melhoria da Qualidade de Software. Discutiu-se sobre o impacto econômico, social e na vida humana promovido por falhas histórias de sistemas de software. As definições de verificação e validação de software foram retomadas e os alunos foram ensinados a construir *checklists* para inspecionar sistemas de software. Ao final da aula, os estudantes tiveram que analisar uma especificação de um programa e escrever casos de teste capazes de fazer a cobertura do programa sob teste. Após a turma redigir os casos de teste, eles foram corrigidos de modo verbal pela professora.

Na terceira aula, os conceitos básicos e termos de jargão da área de teste foram retomados. Também houve uma discussão sobre aspectos associados a definição de conjuntos de casos de teste e os níveis de teste. Na sequência, a existência de procedimentos sistemáticos para condução da atividade e geração de casos de teste foi exteriorizada. Antes da aula ser finalizada, foram disponibilizados 20 minutos para os estudantes responder o pré-teste sobre teste estrutural e critérios Todos-Nós e Todos-Arcos. A turma foi encorajada a estudar os materiais didáticos disponibilizados no ambiente virtual de aprendizagem Moodle, a assistir aos vídeos e fazer anotações sobre as dúvidas que afloraram durante o estudo "antes da aula".

Logo no início da quarta aula, a professora da disciplina dedicou 30 minutos para solucionar as dúvidas dos alunos. Em um primeiro momento, poucos alunos começaram a fazer perguntas. Acredita-se que eles não estavam seguros para formular as questões. Para aguçar a turma a formular questionamento com viés à resolução de dúvidas, a professora começou a perguntar aos alunos se eles haviam entendido informações pontuais, como a geração do grafo de fluxo de controle e a derivação dos requisitos de teste pra cada critério. A partir desses comentários, observou-se que os alunos começaram formular questionamentos. Após a resolução de todas as dúvidas dos participantes, o teste intermediário sobre o conteúdo estudando foi aplicado. Similarmente ao pré-teste, 20 minutos foram disponibilizados para a resolução do mesmo.

Na sequência ao preenchimento do teste intermediário, os alunos foram convidados a formar equipes para realizar uma atividade prática sobre a aplicação dos critérios Todos-Nós e Todos-Arcos. A professora disponibilizou-se para solucionar as dúvidas dos alunos durante a realização da atividade. Os estudantes foram notificados que eles poderiam fazer questionamentos e que a atividade tinha a finalidade de estimular uma melhor compreensão sobre o conteúdo. Em seguida foram disponibilizados guias para o uso das ferramentas EclEmma Java Code Coverage, JUnit, ControlFlowGraph Generator. Os alunos foram advertidos que precisariam se instruir sobre as ferramentas porque iriam manuseá-las em uma atividade prática na aula seguinte.

A quinta aula foi ministrada em um laboratório, em que foram disponibilizados computadores e máquinas virtuais preparadas com ferramentas de apoio à automatização dos critérios baseados em fluxo de controle. No início dessa aula, com propósito de obter *feedback* dos alunos em relação ao aprendizado a partir do modelo *flipped classroom*, o questionário baseado no instrumento definido por He *et al.* (2016) foi aplicado. Na sequência, a descrição sobre a atividade que eles deveriam realizar foi entregue à turma. Um breve treinamento sobre o manuseio e a implementação de casos de teste na ferramenta EclEmma Java Code Coverage foi fornecido. Para o contexto dessa atividade, os alunos foram notificados que deveriam realizá-la de maneira individual, podendo consultar livremente os tutoriais disponibilizados e ao final entregar um relatório, de acordo com o que está previsto pela abordagem FlipClass-STT.

O começo da sexta aula foi destinado à resolução de dúvidas sobre os critérios Todos-Nós e Todos-Arcos. Os alunos deveriam apresentar as suas incertezas sobre o conteúdo e a respeito da atividade prática em laboratório. Após 20 minutos de discussão, o pós-teste foi aplicado. Na sequência, houve uma revisão do conteúdo. Faltando exatamente 30 minutos para a aula ser finalizada, houve a aplicação do pré-teste sobre critérios baseados em fluxo de dados. Na ocasião, os alunos foram alertados sobre a necessidade de estudar o conteúdo disponibilizado no Moodle e se preparar para a próxima aula, anotando dúvidas.

A sétima aula da disciplina ocorreu de maneira similar a quarta aula. Inicialmente, os alunos tiveram um momento dedicado a resolução de dúvidas, na sequência, houve a aplicação do teste intermediário e a realização de uma atividade em duplas com supervisão do professor. Além do conteúdo ser diferente, o tempo para realização da atividade foi diferente. Ao contrário da quarta aula que teve 30 minutos dedicado a resolução de dúvidas, nesta aula foram dedicados apenas 20 minutos, uma vez que os alunos foram pragmáticos e rapidamente apresentaram suas dúvidas. A professora respondeu as dúvidas da turma e instantaneamente aplicou o teste intermediário. Esse teste de conhecimento consumiu 30 minutos do tempo total da aula. Antes da aula chegar ao fim, os alunos receberam instruções sobre o tutorial disponibilizado no Moodle a respeito da ferramenta de teste

Baduino.

No início da oitava aula, os alunos foram convidados a emitir um *feedback* sobre as suas percepções a respeito do modelo *flipped classroom*. Na sequência, houve um momento dedicado a uma preleção sobre uso da ferramenta Baduino, o enunciado da atividade prática foi disponibilizado e a mesma deveria ser realizada de maneira individual pelos estudantes. Ao final da aula, os alunos deveriam entregar a atividade. Durante a realização da atividade, alguns estudantes relataram problemas com o manuseio da ferramenta e não conseguiram executar os casos de teste criados. Foi observado que os alunos não alteraram a versão do JUnit para o projeto que estava sendo trabalhado. Além disso, alguns estudantes atualizaram a versão do Java, inviabilizando a realização da atividade. Diante disso, os alunos que tiveram problemas com o uso da ferramenta foram notificados que poderiam finalizar a atividade "depois da aula" e entregar o relatório até a próxima aula.

Antes da resolução de dúvida previstas para o início da nona semana, os alunos relataram que tiveram dificuldades para usar a ferramenta Baduíno, pois apesar de utilizarem máquinas virtuais alguns não conseguiram realizar a atividade por problemas com atualizações da ferramenta. Por consequência, a atividade não contabilizou nota para a disciplina e foi considerada apenas como uma experiência de aplicação. Nesse sentido, outras ferramentas que oferecem suporte à condução automatizada da atividade (*e.g.* JaBUTi) foram apresentadas. Em seguimento, houve a aplicação do pós-teste, um momento para revisão do conteúdo associado aos Critérios baseados em fluxo de dados e a aplicação do pré-teste sobre a técnica de teste baseado em defeito e o critério teste de mutação. Após a entrega dos questionários de conhecimento, os alunos receberam instruções sobre atividades que deveriam realizar, com o intuito de se preparar para a próxima aula invertida.

A décima aula foi dedicada ao ensino de teste de mutação. Inicialmente, os alunos tiveram 10 minutos para solucionar as dúvidas sobre a técnica e o critério. Na sequência, eles responderam as questões que constituem o teste intermediário e realizaram uma atividade em dupla. A professora acompanhou a resolução da atividade, solucionou as dúvidas e apresentou ao final da atividade a sua resolução. Diferentemente das aulas sobre os critérios de teste estrutural, em que a atividade em equipe foi resolvida somente na aula de revisão, a professora da disciplina conseguiu apresentar uma resolução para essa atividade. Ao final da aula, o material sobre a ferramenta Proteum foi indicado.

No início da décima primeira aula, os alunos foram convidados a emitir um *feedback* sobre o aprendizado de teste de mutação. Dando continuidade, eles tiveram que manusear a ferramenta Proteum. Houve uma breve explicação sobre a ferramenta. Assim, os alunos que não tiveram tempo para aprender sobre a ferramenta, tiveram a oportunidade de aprender. Na sequência, o enunciado da atividade foi disponibilizado. Os alunos

deveriam gerar os programas mutantes, elaborar casos de teste para "matar" os mutantes e identificar mutantes equivalentes. Ao final da aula, os alunos foram convidados a realizar o pré-teste sobre o teste funcional e estudar os materiais disponibilizados no Moodle.

A décima segunda aula foi dividida em duas partes. Primeiramente, as atividades práticas realizadas ao longo da disciplina foram retomadas, visando reforçar o conhecimento dos alunos. Na segunda parte, houve a resolução de dúvidas sobre teste funcional e a aplicação do teste intermediário. Na décima terceira aula, os alunos foram convidados a formar grupos e realizar uma atividade sobre a identificação de requisitos de teste e a elaborar o conjunto de casos de teste a partir dos critérios de teste funcional. A professora esteve presente, solucionado dúvidas, auxiliando os alunos a realizarem a atividade. Ao final dessa aula, os alunos participaram do pós-teste sobre os conteúdos da unidade de teste baseado em defeitos.

Na décima quarta aula os alunos tiveram que emitir o *feedback* sobre o aprendizado de teste funcional a partir do modelo *flipped classroom* e realizar uma atividade prática sem apoio de ferramentas de teste, aplicando critérios de teste funcional. Na sequência, os alunos foram avisados sobre as correções realizadas e notas obtidas ao longo da disciplina. Por fim, responderam ao último teste de conhecimento (pós-teste) previsto na disciplina.

A última aula da disciplina foi dedicada a revisão de conteúdos para alunos que não conseguiram atingir as notas mínimas (50%) durante a realização das atividades realizadas ao longo da disciplina. Salienta-se que as atividades intituladas pré-teste não foram consideradas como atividades avaliativas da disciplina. Tampouco as atividades realizadas em equipe com apoio do professor. Todas foram identificadas como atividades de participação. Os resultados obtidos ao longo da disciplina, coletados por meio dos testes de conhecimento e *feedback* estão descritos na próxima seção.

5.5 Resultados

Ao final da experiência obtida com a disciplina, os dados coletados foram analisados. Optou-se por não utilizar técnicas de estatística descrita ou inferencial, uma vez que o grupo de alunos que se matriculou na disciplina e a frequentou durante todo o semestre foi equivalente a 12 indivíduos. Nesse sentido, a fim de expressar os resultados, utilizou-se somente as porcentagens das respostas.

Durante a experiência, os alunos precisaram classificar os recursos educacionais que eles tinham preferência. Cada estudante deveria analisar uma lista de opções e atribuir uma pontuação para cada item disponível na lista. Como haviam doze itens, eles deveriam atribuir uma nota que varia entre 0 a 11 pontos, atribuindo o valor zero para o item com menor prioridade e onze para o item favorito. Esses sujeitos foram notificados que cada item deveria receber uma pontuação e que eles não deveriam dar a mesma pontuação para

dois ou mais recursos.

Durante a análise, foi definida uma equação para ordenar os recursos educacionais em conformidade com as predileções da turma. Com base na pontuação que cada item obteve, foi feita uma média da pontuação por item, colocando a pontuação na escala original (*i.e.*, 0 a 11 pontos). Por fim, foi feito um *ranking* dos itens. O cálculo utilizado para identificar a pontuação atribuída pelo grupo a cada recurso educacional é dado pela seguinte equação:

$$Item = \sum_{i=1}^{N} x_{(\mathrm{i})}$$

em que:

x = representa o escore que o aluno atribuiu ao item, isto é x = {0, 1, 2, ..., 11};

N = representa o número total de resposta obtidas em cada um dos itens;

i = representa os estudantes, isto é, i = {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_n$}.

A partir da identificação da pontuação média atribuída pelos alunos aos recursos educacionais, eles foram classificados em ordem decrescente, do recurso que recebeu a maior pontuação ao que recebeu a menor pontuação pela turma.

As respostas dos estudantes nos testes de conhecimento foram examinadas. Nessa etapa, foram considerados apenas os acertos dos alunos que participaram em todos os testes de conhecimento por tópico da abordagem. Os testes de conhecimento foram corrigidos e a partir da correção foi calculada a frequência de acerto que o aluno obteve em cada teste. Para calculá-la, foi considerada a seguinte equação:

$$F_{(\mathrm{i})} = \frac{n_{(\mathrm{i})}}{n_{(\mathrm{TOTAL})}}$$

em que:

F = representa a frequência de acertos;

n = representa o número de questões que o aluno acertou;

i = representa o aluno, isto é, {aluno $_1$, aluno $_2$, aluno $_3$, ..., aluno $_n$};

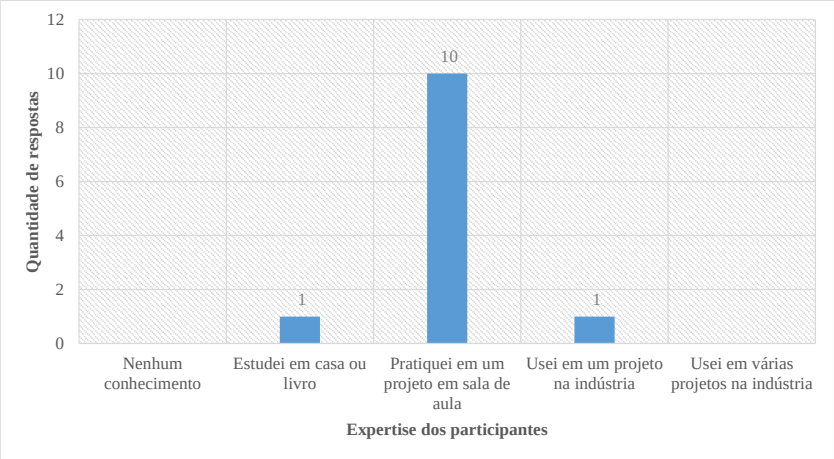
$n_{(\mathrm{TOTAL})}$ = representa o número de questões presentes nos testes de conhecimento.

Na sequência, explorou-se a representação gráfica das taxas de acertos por intermédio de gráficos do tipo *box plot*. Após a análise, os *feedbacks* emitidos pelos alunos durante o estudo foram examinados. De maneira similar a análise anterior, foram considerados os *feedbacks* emitidos por alunos que participaram de todas as atividades propostas para cada um dos tópicos ensinados.

5.5.1 Expertise dos estudantes

Os doze participantes responderam ao formulário de caracterização. Na primeira questão (Figura 35), cada participante teve que avaliar o seu grau de conhecimento na área de engenharia de software. Com base nisso, foi possível constatar que todos os alunos autodeclararam que possuem alguma *expertise* na área de engenharia de software. Dentre os doze estudantes, dez mencionaram que já haviam participado de um projeto em sala de aula. Em particular, um aluno mencionou que somente havia estudado, mas não participou de projeto da área. Outro participante mencionou que já havia empregado o conhecimento obtido na disciplina de engenharia de software em um projeto na indústria.

Figura 35 – Grau de conhecimento dos participantes em Engenharia de Software

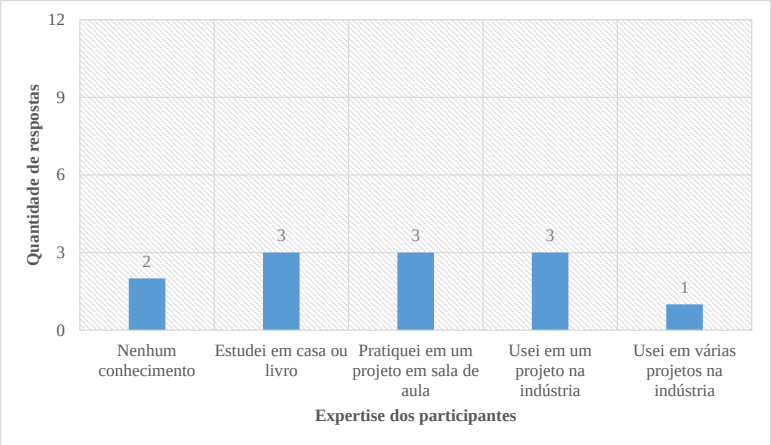


Fonte: Elaborada pelo autor.

A segunda pergunta do questionário de caracterização pediu para os alunos indicarem o quanto eles tinham de conhecimento sobre a área de teste de software. As respostas obtidas foram divergentes (Figura 36). Dois estudantes revelaram que não possuiam nenhum tipo de conhecimento na área. Por outro lado, três alunos mencionaram que já haviam estudado o conteúdo. Outros (três) declararam que já haviam participado de pelo menos um projeto em disciplinas. Três alunos especificaram que haviam participado de um projeto da indústria. Por fim, um aluno indicou que já havia participado de vários projetos na indústria.

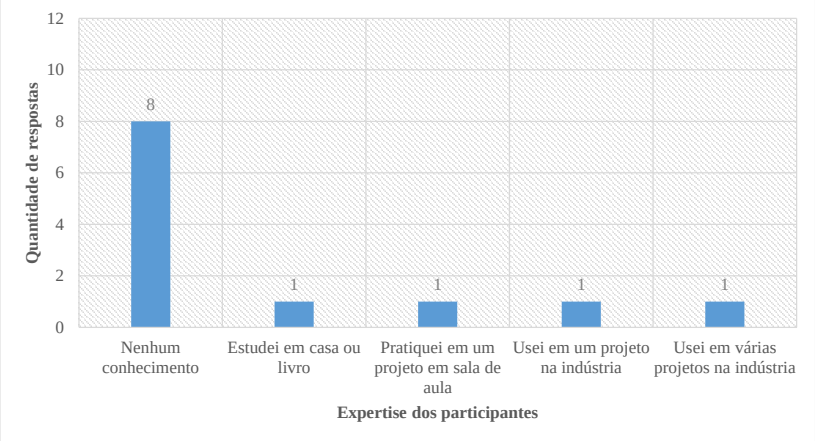
A partir da segunda questão, foram feitas perguntas específicas sobre o conhecimento relacionados aos critérios de teste de software, os quais são contemplados pela abordagem FlipClass-STT. Nessa perspectiva, a terceira questão aborda a temática de teste funcional (Figura 37). Constatou-se que oito alunos não possuíam conhecimento/experiência sobre os critérios da técnica de teste funcional. Um aluno revelou que já havia estudado o conteúdo. Outro estudante mencionou que já havia participado de projetos em sala de aula. Um aluno declarou que já havia usado os critérios em um projeto na indústria e um aluno alegou que havia testado vários projetos de software na indústria.

Figura 36 – Grau de conhecimento dos participantes em Teste de Software



Fonte: Elaborada pelo autor.

Figura 37 – Experiência dos estudantes em utilizar os critérios de Teste Funcional em projetos de software

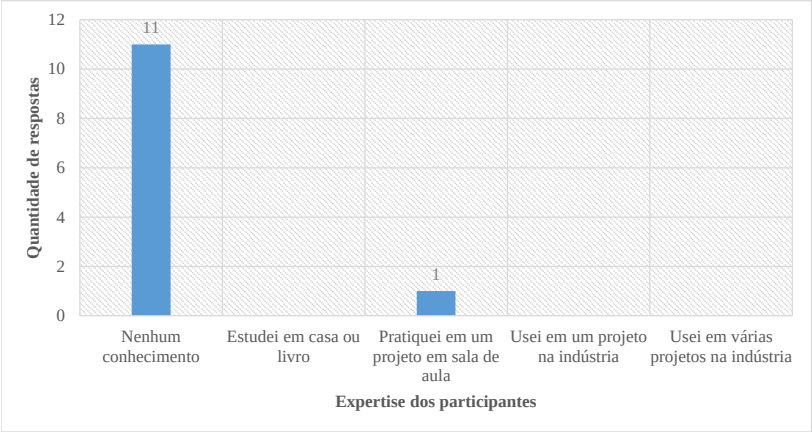


Fonte: Elaborada pelo autor.

Durante a quarta questão, os alunos foram interrogados sobre suas experiências em aplicar critérios de teste estrutural em projetos de software. Constatou-se que somente um estudante possuía conhecimento sobre o critério, adquirido por intermédio de uma disciplina. O restante dos estudantes não possuía conhecimento/experiência sobre a técnica e critérios estruturais. A Figura 38 representa os resultados obtidos.

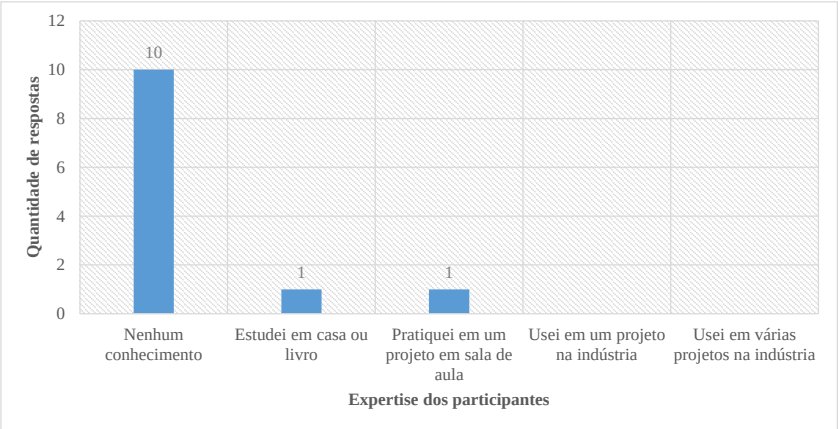
Na quinta questão (Figura 39), cada aluno deveria fazer uma autoavaliação sobre a experiência em aplicar o critério de teste de mutação. Como resultado, dez estudantes mencionaram que não possuíam experiência/conhecimento a respeito do critério. Dois alunos declararam que já haviam estudado o critério, mas somente um deles informou que havia praticado em um projeto em sala de aula.

Figura 38 – Experiência dos estudantes em utilizar os critérios de Teste Estrutural em projetos de software



Fonte: Elaborada pelo autor.

Figura 39 – Experiência dos estudantes em utilizar o critério de Teste de Mutação em projetos de software



Fonte: Elaborada pelo autor.

Por fim, a última questão, por ser opcional e descritiva, foi respondida por dois participantes que relataram que já possuíam experiência na área. Um deles relatou que estava estagiando há aproximadamente 8 meses na área de teste de software. De acordo com o aluno, ele trabalha com teste de projetos de software *web* e *mobile*. Ele descreve que utiliza "técnicas" mais simples como validação de campos de formulário, funcionalidade de botões e análise do valor limite.

O outro aluno comunicou que já havia trabalhado com automatização de teste unitários em aplicações *web*, sistemas *batch* e de processamento paralelo. Ele não soube indicar quais técnicas foram utilizadas, descrevendo que eles não utilizam critérios 'formalmente estabelecidos' pelo meio acadêmico, porque evitavam complexidade e 'testes em excesso'.

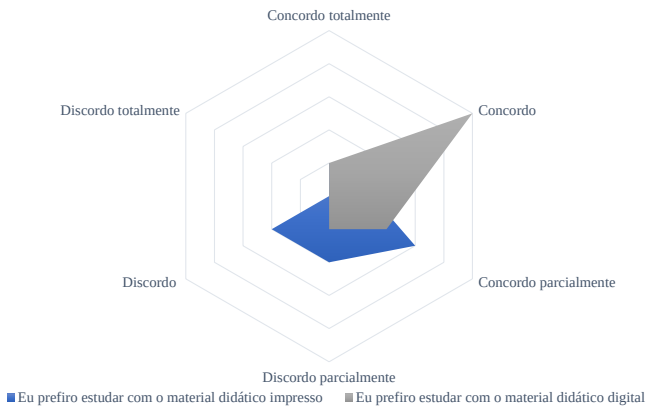
Com base na análise de caracterização, contatou-se que a maioria dos alunos já havia obtido algum contato prévio com a área de teste de software. Alguns obtiveram experiência em critérios de teste durante as aulas do curso e outros por meio de trabalhos realizados na indústria. Observou-se que os alunos que possuíam experiência com teste na indústria, em geral, não utilizavam os critérios de teste para sistematizar a identificação do conjunto de casos de teste. Um dos alunos comentou que a empresa onde ele trabalha busca evitar 'teste em excesso', dando a entender que a organização onde ele trabalha possivelmente não vem conduzindo a atividade de maneira sistemática.

5.5.2 Formato de Recursos Educacionais

Com a finalidade de compreender melhor os interesses dos estudantes por determinados formatos de recursos educacionais, duas assertivas foram definidas. Uma delas buscou averiguar a inclinação dos mesmos por materiais didáticos impressos e a outra o gosto por materiais didáticos digitais. Em relação ao formato impresso, dentre os doze estudantes, quatro mencionaram que não prefere estudar com o material didático impresso. Esses quatro participantes ao serem questionados se preferem estudar com material didático digital, assinalaram opções que variam entre concordo e concordo totalmente, revelando interesse no material digital.

As concordâncias dos participantes pelos diferentes formatos de materiais didáticos foram comparadas. O gráfico radar foi utilizado para mostrar as diferentes séries de dados (Figura 40). Para construí-lo, definiu-se que uma variável seria o material didático impresso, enquanto que a outra variável seria o material didático digital. Com base nisso, foi possível constar que a concordância é maior para a variável material didático digital. Nesse sentido, os dados apontam que há uma tendência entre os alunos em preferir materiais didáticos digitais do que materiais didáticos impressos.

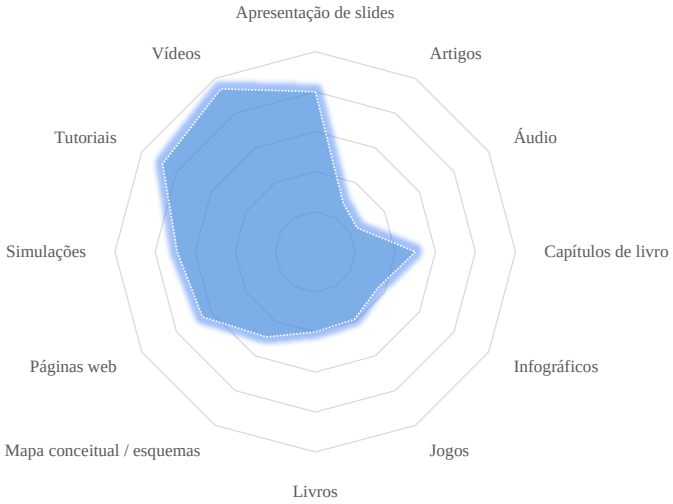
Figura 40 – Comparação entre a preferência por materiais didáticos digitais e materiais didáticos impressos



Fonte: Elaborada pelo autor.

Após o cálculo para classificar a preferência da turma de alunos por materiais didáticos, constatou-se que de modo geral a turma prefere estudar por meio de vídeos. O recurso educacional que a turma menos gosta é o áudio (ou *podcast*). O nível de preferência da turma por materiais educacionais em formato de vídeos é quase quatro vezes maior que o áudio. Para tornar a visualização dos dados mais simples, foi construído o gráfico radar (Figura 41), o qual demonstra a existência de uma tendência pelos seguintes recursos educacionais: vídeos, tutoriais, apresentação de slides, simulações e páginas *web*.

Figura 41 – Tendência por recursos educacionais



Fonte: Elaborada pelo autor.

Após a análise dos materiais, foi possível perceber que os materiais didáticos que os alunos da turma preferem são equivalentes aos recursos educacionais que compõem a abordagem. Nesse sentido, de modo geral, os possíveis problemas associados ao descontentamento com os materiais didáticos, em síntese, foram minimizados, mesmo que de forma não intencional.

Concluída a análise de preferências por recursos educacionais, considerando que houve um acompanhamento do aprendizado ao longo do semestre, por meio dos testes de conhecimento, foi possível ter uma visão geral sobre o aprendizado dos estudantes. Ressalta-se que para o contexto desse relato de experiência, hipóteses, questões de pesquisa ou métricas para testar essas hipóteses não foram definidas preliminarmente. Assim, durante a análise do aprendizado foi considerado apenas a frequência de acertos dos estudantes nos testes de conhecimento.

5.5.3 Aprendizado

Essa seção apresenta os resultados associados ao aprendizado dos estudantes em cada um dos tópicos que são contemplados pela abordagem: (i) critérios baseados em

fluxo de controle; (ii) critérios baseados em fluxo de dados; (iii) teste de mutação; (iv) particionamento em classes de equivalência e análise do valor limite.

Aprendizado de Teste Estrutural – Critérios baseados em Fluxo de Controle

O gráfico da Figura Figura 42 apresenta os resultados a respeito do aprendizado em relação ao teste estrutural. Esses dados proveem dos testes realizados por nove estudantes. É possível observar que o efeito da instrução fez com que a frequência de acertos no teste intermediário aumentasse em relação ao pré-teste.

Os dados do pós-teste aplicado após a realização do conjunto de atividades, demonstram que a taxa de acertos continuou superior aos dados providos do pré-teste, indicando que os alunos aprenderam a partir das instruções e atividades realizadas. Contudo, a realização do conjunto de atividades somente manteve os resultados obtidos durante o teste intermediário, não viabilizando o aumento sob a frequência de acertos.

Observou-se que a medida de tendência central aumentou durante a instrução, uma vez que o pré-teste obteve 38%, enquanto que o teste intermediário obteve uma frequência equivalente a 60%. A medida de tendência central relacionada ao pós-teste, por sua vez, diminui para 50% no pós-teste. Contudo, percebe-se que os valores da frequência de acertos estão menos dispersos no pós-teste do que no teste intermediário.

Acredita-se que as atividades práticas não promoveram efeitos tão bons quanto a instrução realizada. Percebeu-se também que durante o pós-teste alguns alunos acertaram mais questões dos testes de conhecimento que nas aplicações anteriores, uma vez que o limite superior do *box plot* associado ao pós-teste é maior que o limite superior dos outros *box plot*.

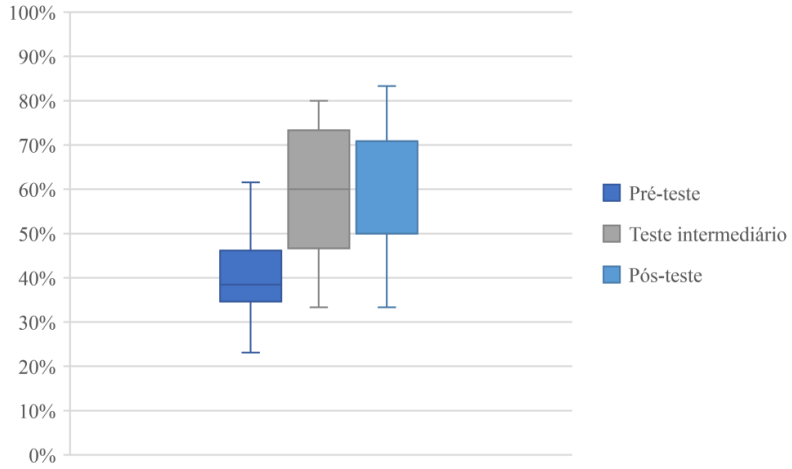
De uma maneira geral, a frequência de acertos nos testes de conhecimento para os critérios baseados em fluxo de controle revela que os alunos conseguiram aprender o conteúdo. Apesar da maioria dos estudantes não conquistar pelo menos 75% de acertos no pós-teste, eles conseguiram atingir o mínimo que é exigido na disciplina, ou seja, 50%.

Os alunos apresentaram um *feedback* com a suas percepções a respeito do modelo *flipped classroom*. Em relação à primeira questão, dentre os nove estudantes, dois alunos manifestaram desinteresse em estudar o conteúdo através deste modelo. Um aluno mencionou que ainda não tinha compreendido o funcionamento, mas estava com grandes expectativas.

No tocante a segunda questão, alguns problemas enfrentados pelos alunos foram mencionados. Um aluno relatou que para ele o modelo trazia novos desafios, porque precisaria conciliar o tempo de estudo com as tarefas do seu trabalho, atividades de disciplinas e do estágio. Outro aluno declarou que acreditava que precisaria de mais experiência para conseguir avaliar o modelo e consequentemente apresentar sugestões de melhoria.

Um outro participante mencionou que acreditava que o modelo aumentaria a sua carga de trabalho. Os demais estudantes não apresentaram sugestões e/ou recomendações. O Quadro 3 apresenta todas as respostas emitidas pelos alunos.

Figura 42 – Frequência de acertos nos testes de conhecimento sobre Critérios baseados em Fluxo de Controle



Fonte: Elaborada pelo autor.

Quadro 3 – Percepções dos estudantes que estudaram os Critérios baseados em Fluxo de Controle com *Flipped Classroom*

	Você gostou ou não do modelo *Flipped Classroom*? Por que?	**Qual é a sua principal reclamação sobre esta experiência e quais são suas recomendações para nós fazermos melhorias?**
1	Gostei.	Nenhuma reclamação/recomendação.
2	Acho bastante interessante o conceito, mas ainda não vi funcionar. Estou com bastante expectativa nessa matéria para ver como vai acontecer o ensino, pois creio que seja uma boa forma de aprender.	Creio que o maior problema é conciliar o tempo disponível em casa para assistir os vídeos, levando em conta que tenho estágio, aulas e algumas atividades complementares, além dos trabalhos das outras matérias.
3	Eu gostei, porque dessa forma eu posso ir para a aula preparada.	Eu acho que seria interessante fazer uma revisão do conteúdo ao final da aula.
4	Gostei, pois estimula o estudante a ver materiais e estudar o assunto.	Nenhuma reclamação ou recomendação para melhorias
5	Não. Eu prefiro a aula tradicional.	Esse modelo, na minha opinião, aumenta muito a carga horária da disciplina.
6	Sim, motiva o aluno a estudar o conteúdo fora da sala de aula.	Acho que é muito cedo para avaliar o aluno.
7	Eu gostei, mas acho que prefiro as aulas tradicionais para entender melhor minhas dúvidas.	<< o aluno deixou a questão em branco >>
8	Gosto, principalmente quando é mesclado com algumas aulas tradicionais.	Não tenho reclamações ou sugestões de melhoria.
9	Eu prefiro aulas tradicionais porque não tenho tempo para acessar o material.	<< o aluno deixou a questão em branco >>

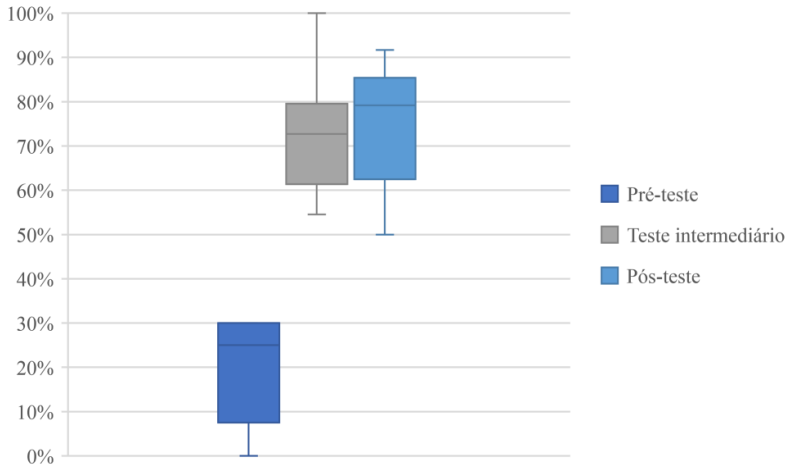
Fonte: Elaborada pelo autor.

Aprendizado de Teste Estrutural – Critérios baseados em Fluxo de Dados

No que diz respeito aos testes de conhecimento relacionado aos critérios baseados em fluxo de dados, durante a análise foram considerados os dados de somente seis estudantes, uma vez que eles participaram de todas as atividades associadas ao aprendizado desse critério. A partir disso, foi possível observar uma ampla variação entre a frequência de acertos dos alunos no pré-teste com os acertos no teste intermediário. Constatou-se também que a variação entre os dados associados ao pré-teste é maior que a variação nos dados do teste intermediário.

Ao comparar a frequência de acertos no teste intermediário com o pós-teste, percebe-se que houve uma maior variação nos dados do pós-teste. Entretanto, a medida de tendência central aumentou, indicando que as atividades podem ter influenciado no número de acertos que estudantes obtiveram. Outro fato importante que é revelado pelo gráfico da Figura 43 é que a partir da instrução, nenhum aluno obteve uma frequência de acertos inferior a 50%.

Figura 43 – Frequência de acertos nos testes de conhecimento sobre Critérios baseados em Fluxo de Dados



Fonte: Elaborada pelo autor.

Nessa etapa do relato, os *feedbacks* dos estudantes também foram coletados. Desse grupo de alunos, somente um mencionou que não apreciou o modelo *flipped classroom*, justificando que ele esquece com mais facilidade o conteúdo que foi aprendido. Os demais alunos mencionaram que estavam apreciando o modelo.

No que diz respeito às sugestões de melhoria, um sujeito mencionou que gostaria de poder realizar atividades mais práticas durante o momento de estudo "antes da aula". Outro estudante, apesar de descrever que gostou de estar aprendendo com *flipped classroom*, relatou que estava sentindo dificuldade em estudar o conteúdo preliminar em virtude do número de tarefas que realizava durante o semestre. O Quadro 4 contém todas

as respostas redigidas pelos alunos.

Quadro 4 – Percepções dos estudantes que estudaram os Critérios baseados em Fluxo de Dados com *Flipped Classroom*

	Você gostou ou não do modelo *Flipped Classroom*? Por que?	**Qual é a sua principal reclamação sobre esta experiência e quais são suas recomendações para nós fazermos melhorias?**
1	Eu gostei. Estou me sentindo mais preparada para a aula	Não tenho reclamação. Eu gostaria de ter mais atividades prática durante o estudo em casa
2	Gostei	Nenhuma
3	Não. Eu continuo acreditando que o conteúdo poderia ser aprendido por completo apenas com aulas expositivas e tarefas em grupo.	Eu acredito que esqueço com mais facilidade o conteúdo
4	Sim, porque estou tendo a oportunidade de estudar sozinho e tirar as dúvidas com a professora	Eu acho que poderiam ser oferecidos mais exemplos nas instruções que precisam assistir
5	Gostei	Não tenho sugestão
6	Gostei, mas eu acho que ainda não estou totalmente com o modelo	Eu trabalho e não tenho muito tempo para estudar. Assim, o modelo dificulta o meu estudo

Fonte: Elaborada pelo autor.

Aprendizado de Teste Baseado em Defeitos – Teste de Mutação

Em relação ao aprendizado de teste de mutação, notou-se que, apesar dos alunos terem mencionado no início da disciplina que não possuíam conhecimento prévio sobre o teste de mutação (Figura 39), eles obtiveram frequências de acertos no pré-teste melhor que o esperado. Todavia, independentemente de conquistarem uma medida de tendência central equivalente a 55% no pré-teste, esses dados foram inferiores aos do teste intermediário e pós-teste (Figura 44).

Acredita-se que os alunos obtiveram uma grande quantidade de acertos no pré-teste porque haviam questões práticas. Nessas questões, os alunos teriam que analisar os programas originais e os programas mutantes, indicando quais operadores de mutação foram alterados. Como as mutações foram relacionadas a operadores relacionais, operadores lógicos e operadores aritméticos, muitos estudantes acertaram essas questões, indicando corretamente a mutação que o programa sofreu.

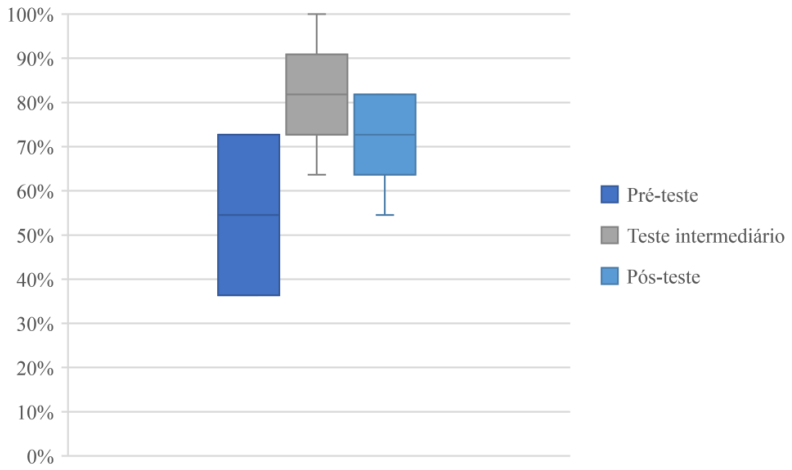
Após a instrução, notou-se que as atividades realizadas pelos alunos “antes da aula”, em conjunto com o momento destinado à resolução de dúvidas, potencializaram o aprendizado. O primeiro quartil relacionado à frequência de acertos no teste intermediário, por exemplo, supera o valor máximo de acertos que os alunos obtiveram no pré-teste. A medida de tendência central, por sua vez, revela que mais de 50% dos participantes obtiveram uma frequência de acertos superior a 80% no teste intermediário .

Similarmente aos resultados apresentados na Seção 5.5.3, notou-se que as atividades práticas realizadas em sala de aula durante o aprendizado de teste de mutação não

contribuíram tanto quanto à instrução. A medida de tendência central do pós-teste apesar de ser superior a medida do pré-teste, é menor que a do teste intermediário (isto é, Pré-teste < Teste intermediário > Pós-teste).

Outro aspecto que pode justificar a redução do número de acertos no pós-teste está associado ao período em que ele foi aplicado. Ocorreu um feriado no período em que esse conteúdo foi ministrado. Assim, é possível que o pós-teste esteja revelando o conhecimento retido pelo aluno ao final do aprendizado, uma vez que haviam se passado três semanas entre a instrução e o pós-teste.

Figura 44 – Frequência de acertos nos testes de conhecimento sobre teste baseado em defeitos – Teste de Mutação



Fonte: Elaborada pelo autor.

Os oito alunos que participaram de todas as atividades de aprendizagem de teste baseado em defeitos emitiram algumas considerações sobre o aprendizado por meio do modelo *flipped classroom* (Quadro 5). Dentre esses alunos, três demostraram descontentamento com o modelo, pois revelaram que gostariam de estudar o conteúdo por meio do modelo tradicional de ensino. Um desses estudantes explanou que preferia ter a teoria sobre o conteúdo em sala de aula e não somente aulas práticas. Apesar desse relato do aluno, conforme descrito na Seção 5.4, aulas mais próximas do tradicional foram intercaladas com as aulas invertidas durante o ensino de cada novo tópico da abordagem FlipClass-STT. Os demais alunos mencionaram que estavam gostando. Salienta-se que um dos estudantes que mencionou que estava gostando revelou que ainda estava com problemas ao se adaptar com o modelo, em virtude de sua carga-horária.

Quadro 5 – Percepções dos estudantes que estudaram Teste de Mutação com *Flipped Classroom*

	Você gostou ou não do modelo *Flipped Classroom*? Por que?	**Qual é a sua principal reclamação sobre esta experiência e quais são suas recomendações para nós fazermos melhorias?**
1	Eu prefiro aulas tradicionais	Gostaria de ter mais teoria em sala de aula, não somente prática
2	Em geral, eu gostei	É difícil estudar com essa metodologia, pois faço outras disciplinas que tem trabalho. Não sei o que poderia fazer, apesar de eu gostar, a disciplina exige uma maior dedicação do que eu havia imaginado
3	Eu gostei, é mais fácil entender o conteúdo da aula se eu estudo antes	O material didático foi um pouco longo
4	Eu gostei	Nenhuma
5	Eu continuo não gostando	Prefiro ter aulas tradicionais
6	Eu gostei, porque eu fico sem dúvidas sobre o conteúdo	Nenhuma
7	Eu gostei, pois o conteúdo estudado não é muito extenso e em contrapartida ganha-se muito tempo em sala de aula	Nenhuma
8	Eu não gostei. Não consigo me adaptar por falta de tempo fora da sala de aula, porque trabalho	Eu acho poderíamos estudar o conteúdo por meio de aulas tradicionais

Fonte: Elaborada pelo autor.

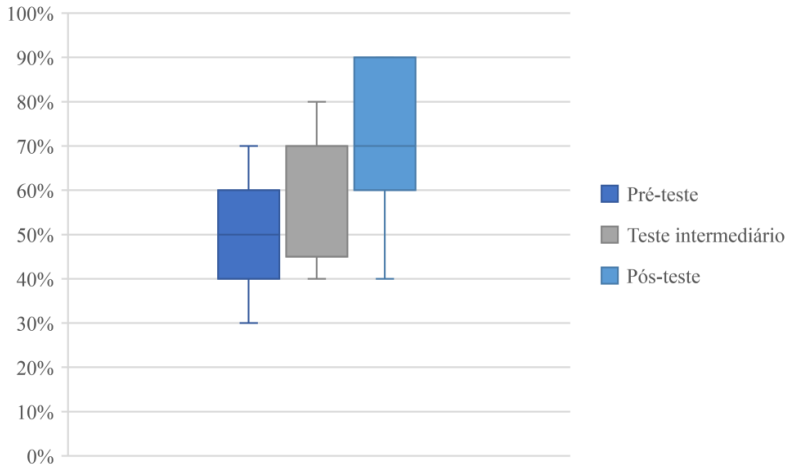
Aprendizado de Teste Funcional – Particionamento em Classes de Equivalência e Análise do Valor Limite

O último tópico ensinado e analisado foi sobre teste funcional e seus dois critérios (particionamento em classes de equivalência e análise do valor limite). Conforme era esperado, houve aumento na frequência de acertos nos testes de conhecimento (Figura 45). A medida que as atividades previstas na abordagem foram realizadas pelos alunos, a frequência de acertos nos testes de conhecimento aumentou.

Ao final da disciplina e do tópico abordado, os alunos emitiram novamente algumas considerações a respeito do aprendizado por meio do modelo *flipped classroom*. Dentre os nove alunos que participaram das atividades previstas para o tópico vigente, três alunos declararam que não gostaram. Dentre os três, dois não descreveram sugestões de melhorias, mas indicaram que preferem o tradicional. Um desses alunos comunicou que ele não rejeitou completamente o modelo, pois achava que poderia aprender com o modelo, mas a disciplina deveria usar *flipped classroom* em uma menor quantidade.

Os seis alunos restantes mencionaram que gostaram de estudar com *flipped classroom*. Cinco deles não declararam sugestões ou emitiram reclamações. O único aluno que gostou e emitiu sugestões, declarou que não havia ficado contente com o formato do material didático utilizado. Todos as respostas emitidas pelos alunos estão expostas no Quadro 6.

Figura 45 – Frequência de acertos nos testes de conhecimento sobre Teste Funcional



Fonte: Elaborada pelo autor.

Quadro 6 – Percepções dos estudantes que estudaram Teste Funcional com Flipped Classroom

	Você gostou ou não do modelo *Flipped Classroom*? Por que?	**Qual é a sua principal reclamação sobre esta experiência e quais são suas recomendações para nós fazermos melhorias?**
1	Eu acho que o modelo exige bastante do aluno. Então, prefiro o tradicional	Nenhuma
2	Gostei	Nenhuma
3	Gostei	Nada
4	Continuo preferindo aulas tradicionais. Portanto, não gostei	Apenas algumas aulas poderiam ser nesse modelo. Eu acho que utilizar esse modelo em todas as aulas deixa a disciplina muito cansativa
5	Gostei	Nada a declarar
6	Gostei	Nenhuma
7	Gostei	Nada a declarar
8	Prefiro o método tradicional	Nenhuma
9	Gostei, pois posso pesquisar antes da aula e se eu tiver alguma dúvida, eu posso levar essa dúvida para a aula e solucionar	Gostaria que o material fosse em outro formato

Fonte: Elaborada pelo autor.

5.6 Lições Aprendidas

Foi possível identificar alguns benefícios com a implantação e algumas dificuldades geradas pela abordagem. Em geral, durante o semestre alguns alunos não conseguiram observar as vantagens de estudar com *flipped classroom*. Em virtude da carga de trabalho gerada pelo momento "antes da aula", estudantes mais conservadores não ficaram contentes e acreditaram que o modelo mais adequado para aprender teste de software é o tradicional, com aulas expositivas.

Os alunos que preferem o modelo tradicional justificaram o descontentamento com o modelo *flipped classroom* em virtude da dificuldade de gerenciar o seu tempo. Foi ob-

servado que os alunos estudavam as demais disciplinas do curso durante a noite e trabalham/estagiavam durante o dia. Eles possuíam uma carga horária em disciplinas grande e ao mesmo tempo que cursavam a disciplina de tópicos, muitos faziam o estágio curricular e o trabalho de conclusão de curso. Outro ponto observado foi que os alunos questionavam a carga horária extra dado que a disciplina não prevê crédito trabalho (crédito por atividade extra-classe).

Com relação ao ponto de vista do professor, foi possível observar que o modelo se torna desafiador pois exige uma nova postura do docente. As aulas deixam de ser completamente previsíveis e sua evolução depende das dúvidas e participação dos estudantes. Um ponto bastante positivo é que surgem alguns questionamentos bem interessantes, que provavelmente não aconteceriam com o modelo tradicional. Um desafio percebido é a vontade do professor explicar algumas atividades quando é desejado que o aluno realize a atividade com base no que estudou antes da aula.

A professora acrescentou que ela acredita que os alunos possuem mais oportunidades para desenvolver o seu aprendizado. Nas aulas, eles terão uma visão mais prática e por meio da prática as dúvidas surgem. Assim, torna-se fundamental o professor em sala de aula para solucioná-las. Além disso, os problemas de uns alunos não são os mesmos dos outros.

Em virtude da disciplina ser ministrada em uma sexta-feira a noite, das 21 horas até as 22 horas e 40 minutos, a quantidade de alunos foi pequena. Aqueles que participaram precisavam cumprir os créditos e se dedicaram para atingir os objetivos previstos no início da disciplina. Contudo, percebeu-se que alguns alunos demonstraram cansaço ao longo do semestre. Inclusive, alguns estudantes utilizaram o número máximo de ausências que a disciplina permitia.

Outro ponto que foi observado está associado com as avaliações. Em decorrência do acompanhamento do aprendizado dos alunos, 12 testes de conhecimento foram aplicados. Essa quantidade foi excessiva e pode ter contribuído com o descontentamento dos alunos. Em um cenário ideal de ensino com a abordagem, espera-se que o professor se apoie nas atividades práticas, não somente nos testes de conhecimento, uma vez que eles não fazem parte da abordagem.

Por fim, a última lição está relacionada à implementação de todos os tópicos. Houve uma reclamação por parte dos estudantes em utilizar quase que predominantemente o modelo *flipped classroom*. Os alunos estão acostumados com o modelo tradicional e isso deve ser levado em conta. Como esse modelo pedagógico não visa uma prática disruptiva, acredita-se que o mais adequado é utilizá-lo com uma frequência menor, intercalando com outros modelos pedagógicos mais próximos do tradicional.

As lições mencionadas no decorrer desta seção buscaram apresentar algumas opi-

niões que são reflexo das observações do pesquisador e dos relatos da professora que implementou a abordagem FlipClass-STT no ensino de teste de software. Espera-se que os professores que a utilizarem, façam relatos sobre as suas experiências com a implementação da abordagem, contribuindo com o seu aperfeiçoamento contínuo. Com base nisso, recomenda-se que sejam discutidas algumas questões como: (i) o que deu certo na experiência em implantar a abordagem FlipClass-STT? ; (ii) o que deu errado? ; (iii) o que melhorar? ; e, por fim, (iv) o que parar de fazer?.

5.7 Apresentação e Empacotamento

Após o planejamento, condução da experiência, análise dos resultados e uma discussão sobre as lições aprendidas, a última etapa do relato de experiência foi apresentar e empacotar os materiais preparados e os dados que foram coletados. O relato de experiência ainda não foi publicado em veículo de divulgação científica. No entanto, espera-se escrever um artigo científico e divulgar a experiência em alguma conferência especializada na área de educação em computação ou mesmo da área de engenharia de software ou teste de software, em alguma trilha dedicada a experiências educacionais.

Todos os materiais desenvolvidos e utilizados no decorrer da experiência e os dados coletados foram disponibilizados em um pacote de laboratório. É possível acessá-los por meio do seguinte endereço: <http://bit.ly/2NqFVkU>. A partir da disponibilização desse pacote de laboratório, espera-se poder contribuir com o ensino e a pesquisa em educação em computação, uma vez que esse material pode ser aproveitado como mecanismo de apoio ao ensino e treinamento de teste de software e utilizado em experimentos relacionados a área, cujo dados provem de testes de conhecimento.

Após a disponibilização dos materiais em um pacote de laboratório, a professora da disciplina de Tópicos Especiais em engenharia de software disponibilizou os dados históricos de versão anteriores da disciplina, em que o mesmo conteúdo foi abordado por meio do modelo tradicional de ensino. A partir da disponibilização dos dados, uma análise histórica de ensino da disciplina foi planejada e conduzida. Essa análise será reportada na próxima seção.

5.8 Paralelo Histórico

Ao final da disciplina, foi feita uma análise dos dados associados ao aprendizado dos estudantes no decorrer da disciplina. Essa análise refere-se às médias finais obtidas pelos alunos, número aprovações, reprovações e a quantidade de alunos que se matricularam e desistiram da disciplina ao longo do semestre (descritos no decorrer da seção como evadidos). Esses dados oferecem uma visão que pode ser utilizada ao fazer um paralelo

com o histórico de ensino da disciplina.

Buscando realizar uma análise comparativa com dados de oferecimentos anteriores dessa disciplina, foram analisados dados da mesma disciplina nos anos de 2011, 2012 e 2014 (ministradas pela mesma professora). Nesses anos, o assunto teste de software foi ensinado, seguindo uma abordagem tradicional e como tarefas para os momentos "depois da aula/antes da aula" os alunos deveriam realizar as atividades práticas da disciplina. Como esperado, outras atividades foram realizadas, e é provável que o perfil da turma seja diferente, então não foi possível comparar as atividades realizadas pelos alunos de maneira individual. Por esses motivos, a visão histórica é abstrata. Ainda assim, a mesma oferece uma visão geral que permite comparar os possíveis efeitos do modelo *flipped classroom* no ensino de teste de software. Algumas informações sobre as disciplinas analisadas são apresentadas na Tabela 35.

Tabela 35 – Detalhes sobre as disciplinas analisadas

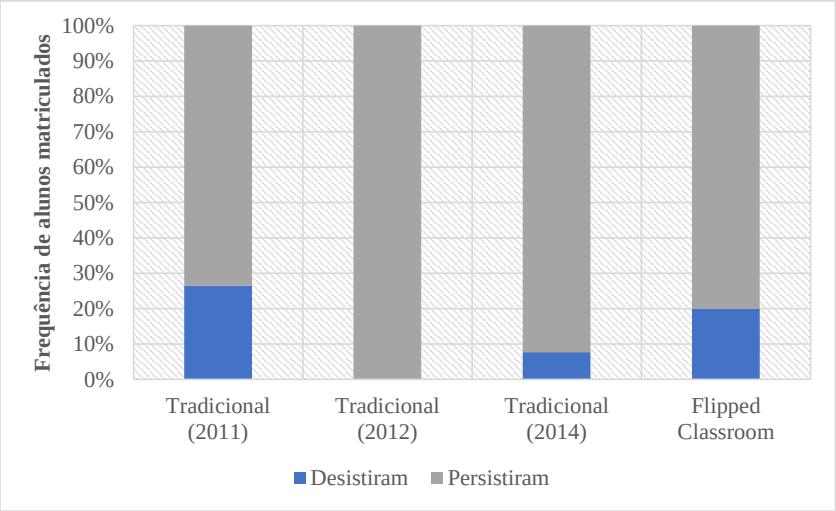
Ano	Número de alunos matriculados	Modelo pedagógico	Dia da semana	Horário
2011	34	Tradicional	Sexta-feira	19hs
2012	14	Tradicional	Quinta-feira	19hs
2014	26	Tradicional	Quinta-feira	19hs
2018	15	*Flipped Classroom*	Sexta-feira	21hs

Fonte: Elaborada pelo autor.

A primeira comparação realizada foi relacionada ao número de alunos que desistiram de cursar a disciplina ao longo do semestre. Observa-se que historicamente a disciplina apresenta uma tendência em obter alunos que se matriculam e desistem. O ápice de desistência ocorreu no ano de 2011, em que aproximadamente 25% dos estudantes desistiram de cursar a disciplina (Figura 46). No decorrer dos anos o índice começou a diminuir. Durante o ensino por meio do modelo *flipped classroom*, em relação aos dados provenientes do ano de 2012 e/ou 2014, observou-se que houve um número maior de desistência. Contudo, a quantidade de alunos que desistiram da disciplina durante o aprendizado de teste de software com *flipped classroom* é inferior ao ano de 2011.

Em razão de ter ocorrido uma diminuição da evasão de alunos nos anos de 2012 e 2014, é possível que o modelo pedagógico *flipped classroom* possa ter afetado o número de alunos que desistiram durante o semestre de 2018. Porém, não é possível chegar a uma conclusão concreta, uma vez que a disciplina é optativa e, é frequente o número de alunos que acabam desistindo das mesmas.

Figura 46 – Frequência de alunos que desistiram da disciplina



Fonte: Elaborada pelo autor.

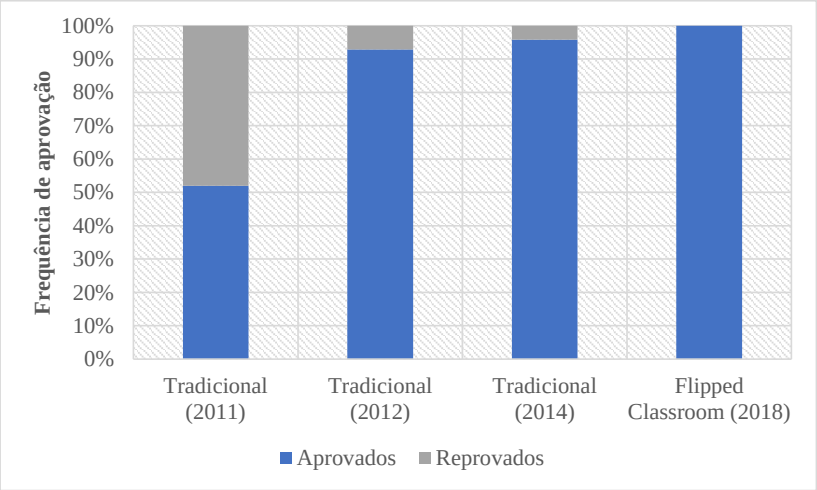
Desconsiderando os dados dos alunos que evadiram, realizou-se uma análise relacionada aos resultados de alunos que reprovaram e aprovaram nos oferecimentos da disciplina. A Figura 47 contém os dados de frequência de aprovação. É possível perceber que o ano crítico foi o de 2011, onde somente 50% dos alunos foram aprovados na disciplina. Em geral, quando teste de software foi ensinado com o modelo tradicional, sempre houveram alunos que não conseguiram atingir os critérios estabelecidos ao longo do semestre, ou seja, não atenderam os objetivos de aprendizagem. Por consequência disso, inevitavelmente acabaram reprovando na disciplina. Com a experiência de *flipped classroom*, nenhum aluno foi reprovado.

Os resultados apresentados na Figura 47 são consequências da dedicação dos alunos durante o aprendizado, uma vez que eles realizaram todas as atividades avaliativas durante a aula. Nos anos anteriores, somente os testes de conhecimento eram realizados em sala de aula. A maioria dos exercícios e atividades eram realizados fora da sala de aula ("antes da aula/depois da aula"), porque o tempo total da disciplina não era suficiente para o ensino do conteúdo e realização das atividades de fixação. Acredita-se que oportunizar o momento da aula para realização de exercícios foi significativo para os alunos, uma vez que com isso, todos que participavam da disciplina realizam as atividades, parcialmente ou totalmente.

A participação dos alunos em sala de aula durante o ensino por meio do modelo *flipped classroom* é importante para o aprendizado efetivo, uma vez que "durante a aula" são promovidas atividades para consolidar a aprendizagem. Nesse sentido, a frequência (presença/ausência) dos participantes durante as aulas foi analisada e comprada com os dados históricos. Para a análise, foram considerados os dados dos alunos que não desistiram da disciplina (Tabela 36). Conforme é possível perceber, a participação dos

alunos durante o aprendizado com *flipped classroom* apresentou uma ampla variação, especialmente se comparado os anos de 2012 e 2014. O ano que obteve uma participação com menor variação foi em 2012.

Figura 47 – Frequência de alunos que foram aprovados/reprovados na disciplina



Fonte: Elaborada pelo autor.

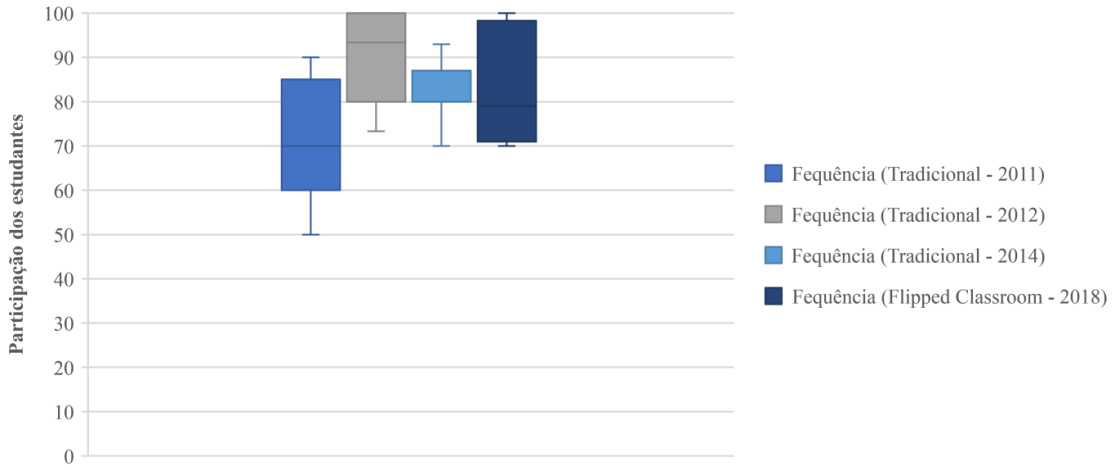
O ano que obteve uma frequência menor foi em 2011. Em 2012 e em 2018, alguns estudantes participaram de todas as aulas da disciplina, uma vez que o valor máximo é equivalente a 100. O valor mínimo de frequência para o ano de 2018 foi de 70, indicando que alguns estudantes não participaram de 30% das aulas práticas. A medida de tendência central indica que a participação maior ocorreu quando a disciplina foi ministrada com meio do modelo tradicional de ensino, em 2012. Nesse período a disciplina foi ministrada em um quinta-feira as 19 horas da noite. Com base na análise, acredita-se que o horário em que a disciplina foi ministrada no ano de 2018 e o dia da semana pode ter contribuído com a participação menor de alguns alunos.

Tabela 36 – Quantidade de alunos não desistentes

ANO	QUANTIDADE DE ALUNOS
2011	25
2012	14
2014	24
2018	12

Fonte: Elaborada pelo autor.

Figura 48 – Frequência dos alunos nas aulas



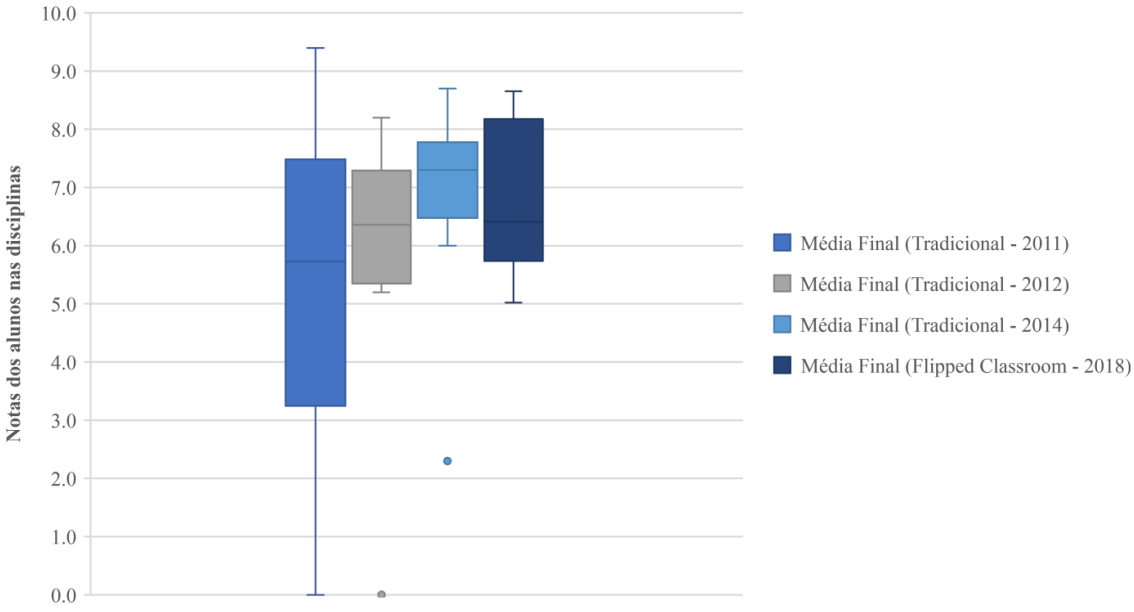
Fonte: Elaborada pelo autor.

A última característica avaliada foi a média final dos participantes nas disciplinas. O ano em que as médias da turma foram menos dispersas foi em 2014. Esse ano também contém a maior medida de tendência central, indicando que os alunos obtiveram melhores notas. Em 2011, alguns estudantes que frequentaram a disciplina não conseguiram obter pontuações suficientes para aprovação na disciplina. Por outro lado, alguns alunos conquistaram notas superiores a 7,5 pontos. Por conta disso, a maior variação de médias finais relaciona-se a 2011.

A medida de tendência central referente as médias finais dos participantes que estudaram com *flipped classroom* foi equivalente a medida correspondente ao ano de 2012. Contudo, ela foi menor que 2014. Esses valores podem estar relacionados com a frequência dos alunos que estudaram por meio de *flipped classroom* nas aulas. Observou-se na Figura 49 que durante o ano de 2018 alguns alunos apresentaram a frequência mínima exigida na disciplina. Como as atividades da disciplina foram realizadas durante a aula, os alunos que faltaram perderam a atividade e, consequentemente, deixaram de pontuar na disciplina.

Ao final da análise, algumas suposições emergiram, uma vez que as comparações realizadas ao longo dessa seção consideraram disciplinas tradicionais ensinadas em dias da semana e horários diferentes dessa disciplina. Além disso, não foi possível ter uma caracterização dos sujeitos que frequentaram a disciplina nos anos de 2011, 2012 e 2014. As suposições que foram concebidas são: (i) os alunos que estudam com *flipped classroom* tendem a desistir da disciplina na mesma frequência que os alunos que estudam com o modelo tradicional; (ii) os alunos que estudam com *flipped classroom* tendem a obter índices de aprovação maiores que os alunos que estudam com o modelo tradicional; (iii) os alunos que estudam com *flipped classroom* tendem a ter uma frequência equivalente aos alunos que estudam com o modelo tradicional; por fim, (iv) a frequência da participação

Figura 49 – Média final dos alunos



Fonte: Elaborada pelo autor.

dos alunos que estudam com o modelo *flipped classroom* influencia nas notas que os alunos obtem na disciplina.

5.9 Considerações Finais

Este capítulo teve o intuito de apresentar um relato de experiência sobre a aplicação da abordagem FlipClass-STT em uma disciplina de teste. Durante o ensino, dados foram coletados com a finalidade de obter uma visão sobre o aprendizado dos estudantes frequente as atividades propostas pela abordagem. Observou-se que os alunos aprenderam e conseguiram aprovação ao final da disciplina. Para complementar, os dados finais da disciplina foram confrontados com dados históricos, cujo conteúdo ministrado foi equivalente ao abordado no decorrer da disciplina. Algumas observações foram realizadas e suposições foram levantadas.

Após a experiência em implantar a abordagem como o modelo, acredita-se que foi possível obter subsídios para responder a segunda questão de pesquisa que esta dissertação se propôs a abordar. Na sequência, com a intenção de contribuir com o aprendizado de teste de software com *flipped classroom*, buscou-se investigar o uso de um agente conversacional como mecanismo de apoio ao *flipped classroom*. Nessa perspectiva, o próximo capítulo abordará uma pesquisa conduzida concomitantemente ao desenvolvimento e aplicação da abordagem FlipClass-STT, a qual envolveu o desenvolvimento, avaliação e estudo sobre um agente conversacional para o ensino de teste de software em conjunto com o modelo *flipped classroom*.

CAPÍTULO

6

O AGENTE CONVERSACIONAL TOB-STT

6.1 Considerações Iniciais

Dentre os momentos previstos pelo modelo *flipped classroom*, o período em que o aluno deve estudar o conteúdo para se preparar para a aula pode ser considerado um dos mais crítico. Para um aluno com histórico de aprendizagem baseada em métodos tradicionais, acostumado a frequentar cursos presenciais e receber instruções em sala de aula, a mudança repentina pode causar estranhamento. É possível que o aluno, por exemplo, comece a acreditar que o professor não tenha a intenção de lhe ensinar o conteúdo. Nesse sentido, visando não criar uma falsa ilusão no estudante e facilitar o aprendizado "antes da aula", discute-se na literatura sobre indicar o material adequado para o aluno estudar.

Conforme descrito no Capítulo 1, também existem alguns relatos de experiência sobre a implementação do modelo *flipped classroom* em que os alunos destacam que têm dificuldade em aprender antes da aula, uma vez que o professor não está presente para solucionar as dúvidas. Ao mesmo tempo que há relatos sobre a falta do professor para resolver as dificuldades dos alunos, as características sobre o conteúdo precisam ser ponderadas, dado que há determinados conteúdos, especialmente conceitos, que oferecem a base para um conteúdo mais acentuado. Caso os conceitos básicos não sejam compreendidos, o aprendizado do aluno pode ficar prejudicado, pois esse não conseguirá progredir no estudo, dirigindo-se para a aula despreparado/sem o conhecimento prévio necessário.

Com base nisso, este capítulo aborda um estudo realizado com o propósito de investigar um mecanismo de apoio ao ensino que pode facilitar o aprendizado dos estudantes durante o momento "antes da aula". O mecanismo em questão consiste em um sistema de software conhecido na literatura como agente conversacional, o qual tem o intuito de solucionar as dúvidas do usuário sobre o conteúdo, emitir sugestões de materiais, exemplificar

situações que podem ser adequadas para a utilização de um dado conhecimento, dentre outros. O estudo em questão utiliza a temática de ensino de teste de software com *flipped classroom* como um caso particular para explorar o potencial dos agentes conversacionais. Assim, busca-se responder a terceira questão de pesquisa abordada nesta dissertação: *"Um agente conversacional é útil para o modelo flipped classroom no contexto de uma aula de teste de software?"*.

Este capítulo encontra-se organizado em seis seções. A Seção 6.2 apresenta uma pesquisa inicial sobre agentes conversacionais para apoiar a educação e a busca por meios de desenvolvimento de agentes. Na Seção 6.3 são discutidos aspectos inerentes à concepção e implementação do agente conversacional TOB-STT para apoiar o ensino de teste de software. Durante a Seção 6.4, uma avaliação preliminar que foi conduzida é apresentada. A Seção 6.5 apresenta os resultados da utilização do TOB-STT como um mecanismo de apoio ao *flipped classroom*. Por fim, são apresentadas algumas considerações finais do capítulo.

6.2 Agentes Conversacionais para Educação

Para compreender a aplicabilidade sobre a disponibilização de um agente conversacional como mecanismo de apoio ao momento de aprendizado "antes da aula" em uma aplicação de *flipped classroom*, em um primeiro momento, foi estabelecida uma revisão de literatura. Em suma, a revisão de literatura foi definida com o propósito de identificar agentes conversacionais disponíveis para a utilização no contexto desta pesquisa. Assim, para apoiar a revisão e ampliar a busca por agentes conversacionais, um mapeamento sistemático foi conduzido.

Durante o planejamento do mapeamento sistemático, algumas reflexões foram realizadas. Dentre elas, ocorreu a suposição de não existirem agentes conversacionais com conhecimento sobre teste de software, livres para utilização. Diante disso, o mapeamento sistemático foi planejado de modo que fosse considerado a revisão sobre os agentes conversacionais desenvolvidos para apoiar práticas educacionais, independentemente de área, em que seria observado se há algum agente para apoiar o ensino de teste. Nesse estudo, buscou-se reconhecer as estratégias (*e.g.* lógica se sistema, linguagens, dentre outras) utilizadas durante a concepção de agentes conversacionais. Assim, em caso do não reconhecimento de um agente conversacional adequado para o contexto deste trabalho, a pesquisa englobaria o desenvolvimento de um protótipo de agente conversacional. A partir disso, o mapeamento sistemático foi planejado, conduzido e reportado, apresentado no Apêndice A.

Ao final do mapeamento sistemático, não foi possível identificar um agente conversacional desenvolvido para ensino de teste de software. Contudo, foi possível identificar

que há uma tendência no desenvolvimento de agentes conversacionais, no que se refere à utilização da lógica de desenvolvimento de sistemas Casamento de Padrões (*Pattern Matching*) e a linguagem AIML. A partir disso, foi pré-estabelecido um agente conversacional. Os detalhes associados ao desenvolvimento do agente conversacional são apresentados na próxima seção.

6.3 Visão Geral do Agente Conversacional

O agente conversacional foi estabelecido para ser capaz de responder as perguntas emitidas por estudantes da área de teste de software. Em primeiro momento, ele consegue responder os questionamentos feitos em formato de texto, no idioma inglês. TOB-STT foi projetado para seguir o formato de software livre, estando disponível para evoluções e manutenções na plataforma GitHub[1].

O propósito do agente conversacional é oferecer apoio ao aprendizado do aluno, possuindo conhecimento a respeito dos principais conceitos, critérios e técnicas de teste de software, especialmente sobre o conteúdo que fazem parte da abordagem FlipClass-STT, de modo que o TOB-STT possa ser utilizado em conjunto com a abordagem. No decorrer desta seção, detalhes técnicos sobre o desenvolvimento do TOB-STT são abordados.

6.3.1 Delimitação do Conhecimento

Na literatura há uma discussão sobre a modelagem do conhecimento de um agente conversacional. Alguns autores que desenvolvem bases de conhecimento para agentes baseados em casamento de padrões e AIML, mencionam sobre a dificuldade em prever quais são as possíveis perguntas que serão emitidas pelo usuário. Como há o casamento entre os padrões (entrada e saída), é necessário estipular as possíveis entradas e oferecer uma ou um conjunto de respostas para as possíveis entradas.

No estudo de Leonhardt *et al.* (2003), os autores descrevem sobre reunir profissionais/professores que possuem *expertise* na área, os quais estão acostumados a solucionar as dúvidas dos alunos, de modo que consigam definir preliminarmente as entradas que precisam ser modeladas e, consequentemente, necessitam estar presentes na base de conhecimento do agente. Recentemente, agentes conversacionais que se baseiam em AIML e casamento de padrões vêm sendo desenvolvidos de maneira similar. Cita-se como exemplo, os estudo de Santos (2009), Aguiar, Tarouco e Reategui (2014), Lucchesi *et al.* (2018), Paschoal, Oliveira e Chicon (2018).

Diante disso, a base de conhecimento do agente TOB-STT foi definida por meio da experiência da equipe envolvida no projeto de pesquisa. Durante a delimitação do

[1] Disponível em: <https://github.com/paschoall/tob-stt>.

conteúdo, foi definido que o desenvolvimento da base de conhecimento do agente seria feito de modo gradativo, começando pelas terminologias de teste. Na sequência, dar-se-ia início as bases associadas aos critérios de teste funcional e estas seriam seguidas por bases relacionadas a critérios de teste estrutural e teste de mutação.

No projeto foi definido que para cada assunto abordado, diferentes tipos de perguntas seriam exploradas, associadas a conceitos, demonstrações e exemplos. Nesse sentido, estabeleceu-se que as questões teóricas relacionadas a conceitos teriam padrões de respostas baseados nas descrições do SWEBOK (BERTOLINO; MARCHETTI, 2014). Para os padrões relacionados a demonstrações de aplicações dos critérios, algumas questões dos testes de conhecimento produzidos no decorrer dos estudos experimentais (Capítulo 3) e ao longo da aplicação da abordagem FlipClass-STT (Capítulo 4) foram selecionadas e adaptadas. Para complementar a base de conhecimento, alguns exemplos de utilização dos critérios durante a atividade de teste foram definidos, tendo como base questões do CTFL[2].

Além das bases de conhecimento relacionados ao conteúdo de teste de software, foi considerada a argumentação feita por Barros e Tedesco (2016) em relação à conversação ser uma atividade estruturada. Assim, para o contexto desta pesquisa, foi considerado que a conversa é dividida em três fases distintas: abertura (ou início da interação), desenvolvimento e fechamento (ou fim da interação). Cada uma dessas fases corresponde ao momento de uma conversa. A abertura corresponde ao momento em que os sujeitos que participam da conversa se apresentam (*i.e.*, aluno e agente conversacional). No desenvolvimento, os sujeitos conversam sobre tópicos associados ao conteúdo. No fechamento da interação, os sujeitos se despedem.

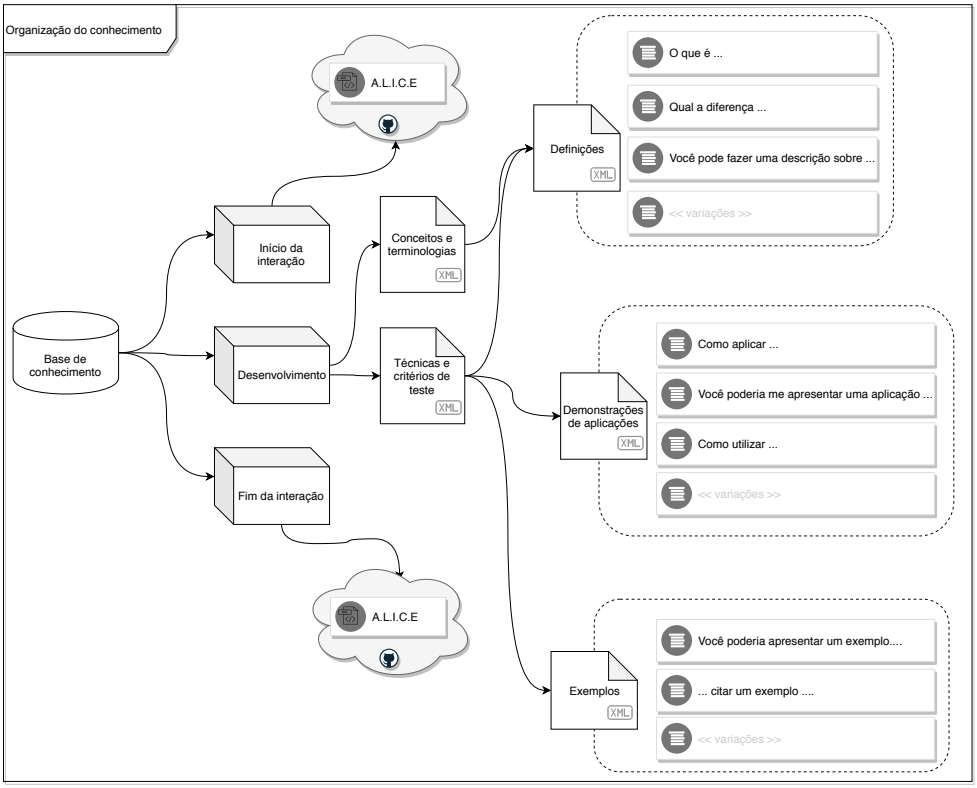
Tendo em vista a construção de um agente capaz de contemplar as três fases da conversação, algumas ações foram tomadas. Assim, buscando minimizar o esforço para produzir diferentes tipos de bases de conhecimento e priorizar as bases associadas ao conteúdo (fase de desenvolvimento), preferiu-se reutilizar bases de conhecimento associadas as fases de início e fim da interação. Para tanto, foi necessário identificar bases de conhecimento que estão disponíveis para reúso, em formato livre e aberto e que já foram utilizadas em projetos similares. Nessa perspectiva, foi estabelecido que as bases de conhecimento originais do projeto A.L.I.C.E (*Artificial Linguistic Internet Computer Entity*) (WALLACE, 2009) seriam adotadas.

Com base nas decisões tomadas para o planejamento da base de conhecimento, foi construída uma representação que evidencia a estrutura da base de conhecimento do agente. A Figura 50 apresenta uma visão geral sobre como ela foi organizada. Nota-se que a base é constituída por um conjunto de diretórios (*e.g.*, ínicio da interação) que contêm diversos arquivos implementados em AIML (*e.g.*, conceitos e terminologias, definições),

[2] Mais informações disponíveis em: <https://www.bstqb.org.br/sobre-ctfl>

cuja estrutura é equivalente a linguagem XML. Um deles inclui o conjunto de arquivos AIML com bases sobre cumprimentos. Outro diretório engloba os arquivos associados ao conteúdo de teste de software. Por fim, o último diretório contém arquivos com bases de conhecimento sobre saudações.

Figura 50 – Estruturação do conhecimento do agente TOB-STT



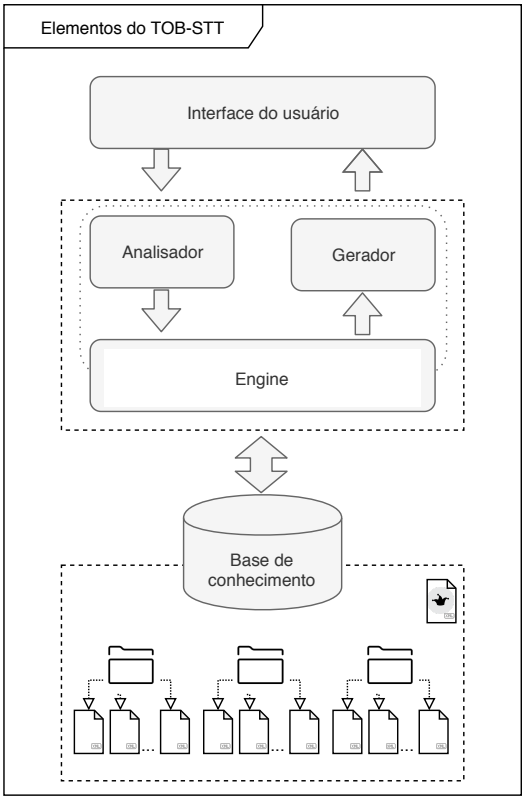
Fonte: Elaborada pelo autor.

6.3.2 Características e Implementação

TOB-STT foi desenvolvido empregando um conjunto de tecnologias específicas para o desenvolvimento de agentes conversacionais. Uma representação dos componentes que constituem o agente é apresentada na Figura 51. TOB-STT é constituído por (i) uma base de conhecimento que armazena todo o conhecimento do agente; (ii) uma interface que é o principal canal de comunicação do usuário com o agente, garantindo a comunicação entre o usuário e o agente; (iii) uma máquina de inferência (ou um interpretador) que tem a função de processar a entrada do usuário (analisador), localizar os dados de resposta em uma base de conhecimento e gerar a saída (gerador).

O processo básico de funcionamento do TOB-STT engloba algumas etapas. O aluno deve interagir com o agente por meio de sua interface gráfica. A interface é constituída por um campo de entrada de dados, no qual o usuário deverá digitar a sua pergunta. A interface também contém uma área destinada às respostas. Nesse sentido, o aluno deve

Figura 51 – Principais elementos que constituem o agente TOB-STT



Fonte: Elaborada pelo autor.

interagir com o TOB-STT digitando as perguntas em linguagem natural no campo específico. O agente irá processar essa entrada e procurar informações em sua base de conhecimento. Se o TOB-STT identificar a resposta em sua base de conhecimento, ele irá retornar a resposta. Por outro lado, se ele não obtiver uma resposta para o questionamento ou comentário feito pelo aluno, ele informará que não possui uma resposta para o assunto, uma vez que ainda está aprendendo, sugerindo ao usuário contar-lhe sobre o que se trata, para poder ter um entendimento melhor.

Tecnologias para Desenvolvimento

A definição das tecnologias e ferramentas para apoiar o desenvolvimento do agente TOB-STT foi baseada em alguns princípios:

- Utilização de tecnologias de software livre, buscando produzir um agente conversacional *open source*. Com isso busca-se estimular a utilização desse recurso educacional, permitindo que o outros pesquisadores e professores possam estudá-lo e adaptá-lo de acordo com as suas necessidades. Além disso, o formato de software livre tem o intuito de estimular o aperfeiçoamento do TOB-STT, de modo que toda a comunidade de educação em engenharia de software se beneficie.

- Reutilização de software, tendo a finalidade de aumentar a qualidade e produtividade no desenvolvimento do agente conversacional. Nessa perspectiva, bases de conhecimento elaboradas na estudo de Wallace (2009) para o agente A.L.I.C.E e disponíveis em formato livre por meio da plataforma GitHub foram recuperadas e reaproveitadas. Em especial, as bases relacionadas às fases de início e fim de interação foram reutilizadas. Durante o reaproveitamento das bases de conhecimento, evitou-se fazer mudanças nas bases, abstendo-se de não comprometer a performance das mesmas.

- Utilização de tecnologias *web*, de modo a facilitar a produção de um agente conversacional facilmente integrável com ambientes virtuais de aprendizagem, repositórios de materiais, sistemas tutores inteligentes e mundos virtuais.

Com base nessas características, a linguagem de programação utilizada foi o PHP (versão 5.4.0), por ser a mesma linguagem utiliza no interpretador de AIML denominado Programa O[3], o qual foi adotado como máquina de inferência do TOB-STT. Além disso, essa linguagem é amplamente utiliza em ambientes virtuais de aprendizagem como o Moodle, permitindo uma maior flexibilização por professores que desejam inseri-lo no contexto de cursos de teste de software, por exemplo, além de ter como característica a portabilidade, robustez e independência sobre o sistema operacional.

A interface, por sua vez, foi implementada tendo como base o interpretador Program O, o qual além oferecer uma máquina de inferência, também possibilita gerenciar as bases de conhecimento do TOB-STT, registrar o histórico das interações e concede uma interface para bate-papo. Essa interface sofreu algumas modificações e foi implementada em HTML (versão 5) com AJAX (*Asynchronous Javascript and XML*, permitindo maior portabilidade ao acesso ao TOB-STT (*e.g.*, seu uso em tablets e notebooks).

Em virtude da utilização do Program O, as bases de conhecimento são gravadas em um banco de dados MySQL que armazena os arquivos AIML usados para formular as respostas do agente, os registros de acesso e conversação de usuários com o TOB-STT, dentre outros dados associados ao gerenciamento do agente.

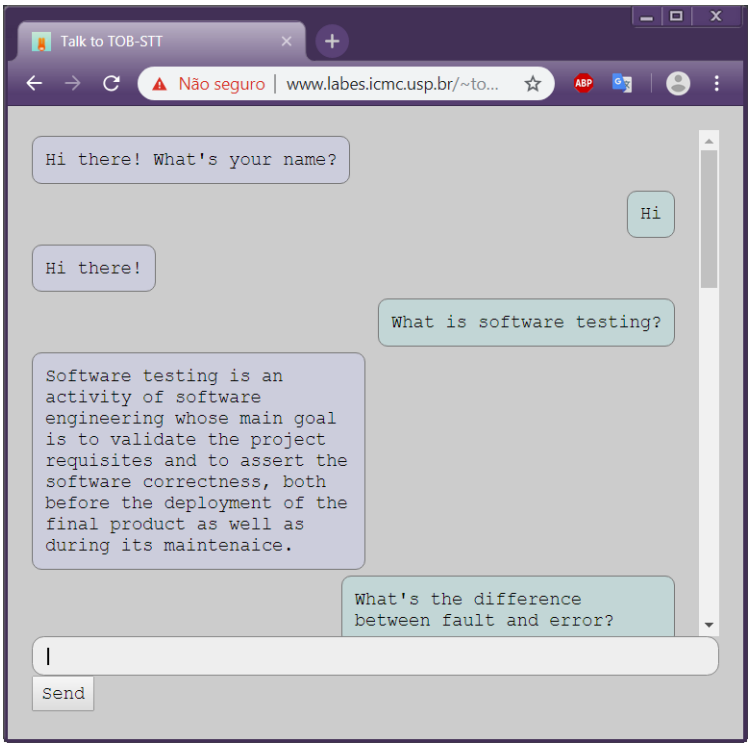
Interface do Agente Conversacional

TOB-STT possui uma página que apresenta o componente de diálogo, isto é, o campo que permite que o usuário digite uma mensagem textual (pergunta ou comentário) e o campo para que o agente conversacional escreva sua resposta. Após o usuário digitar a pergunta, ele deve clicar no botão ”Send”ou teclar ENTER. Salienta-se que a interface foi desenvolvida para ter uma característica de bate-papo, similar a outros sistemas de

[3] Mais informações disponíveis em: <https://github.com/Program-O/Program-O>

software como o WhatsApp, Telegram e o Messenger. Com essa característica, buscou-se concentrar o usuário na interação em linguagem natural, estimulando-o a explorar as respostas fornecidas pelo agente conversacional. A Figura 52 apresenta a tela inicial do agente TOB-STT.

Figura 52 – Interface do TOB-STT



Fonte: Elaborada pelo autor.

O TOB-STT não foi planejado para possuir um avatar para representá-lo. Essa decisão foi tomada em virtude de alguns fatores. Primeiramente, a maioria dos agentes conversacionais para educação identificados no mapeamento sistemático não possuem uma representação gráfica. Outro ponto que contribuiu com a decisão tomada está associado com a necessidade de se fazer um estudo mais profundo sobre representação visual e gênero de agentes conversacionais. Somente com um estudo preliminar é possível definir uma aparência, que não comprometa a percepção do usuário sobre o agente. Além disso, é necessário definir tecnologias de software livre para modelar e desenvolver a aparência.

Base de Conhecimento

As bases que constituem o conhecimento do agente TOB-STT foram implementadas através de diferentes comandos (ou *tags*) do AIML. Além das *tags* mais básicas, foram utilizadas outras que tem a capacidade de criar diálogos bem complexos, buscando oferecer tratamento de informações em quadro níveis linguísticos, a saber: morfossintático, semântico, discursivo e pragmático. Cada um dos tratamentos realizados serão apresentados no

decorrer da seção. Salienta-se que para ilustrar e exemplificar as ações tomadas, visando facilitar o entendimento, os códigos que são exemplos serão apresentados no idioma em português, em decorrência de ser o idioma utilizado no texto da dissertação.

Morfossintático

TOB-STT pode fazer o tratamento morfológico da informação enviada pelo usuário. Para tanto, a base de conhecimento contém *tags* que são capazes de corrigir erros de digitação e padronizar termos a partir de variações que não são necessariamente um erro. Essas *tags* foram utilizadas em uma base destinada à substituição de palavras. A seguir é apresentado um comando que pode ser utilizado para corrigir um erro de digitação: <substitute find="softwre"replace="software"/>. O mesmo comando pode ser utilizado para padronizar variações de digitação como: <substitute find="software test"replace="software testing"/>.

Em uma conversa entre seres humanos, uma pergunta pode ser respondida de diferentes maneiras. Desse modo, com o intuito de simular diferentes formas de resposta para uma mesma pergunta, a linguagem AIML possibilita moldes de réplicas. Nesse sentido, a base de padrões de diálogo constitui-se por unidades que são identificadas pela *tag* <category>. Cada unidade é composta por um padrão de entrada <pattern> e um ou mais moldes de réplica <template>. No Código-fonte 1 é apresentado um exemplo de uma interação em que uma mesma pergunta pode ser respondida de diferentes maneiras. A *tag* <random> contém uma lista (representado pelo comando <li>) de possíveis respostas e uma delas será sorteada aleatoriamente e retornada como resposta para o usuário.

Código-fonte 1 – Exemplos de respostas com uso de recursividade

```
1: <aiml>
2:   <category>
3:     <pattern> o que é teste de software? </pattern>
4:     <template>
5:       <random>
6:         <li> é uma atividade que tem como objetivo revelar a
      presença de defeitos em um software. </li>
7:         <li> é uma atividade que visa demonstrar que um software n
      ão está livre de defeitos. </li>
8:         <li> é uma atividade que faz parte do processo de
      desenvolvimento de software, tendo como finalidade revelar a
      presença de defeitos. </li>
9:         <li> é uma atividade essencial no desenvolvimento do
      software capaz de demonstrar defeitos que estão presentes no
      mesmo. </li>
10:       <random>
```

```
11:     </template>
12:   </category>
13: </aiml>
```

Também foi possível tratar expressões desconhecidas utilizando o comando asterisco "*" dentro do padrão de entrada <pattern>. Assim, quando o usuário questiona o TOB-STT sobre um assunto que não foi previamente definido, ele responde que ainda não possui conhecimento sobre o assunto ou não está conseguindo entender o que o usuário quer dizer.

Semântico

Uma base de descrição que contém todas as características específicas sobre o TOB-STT foi definida. O comando <property> facilitou as descrições sobre a caracterização. Dentre as informações que foram definidas cita-se como exemplo o nome do agente conversacional e data de criação. A seguir um comando é demonstrado como exemplo: <property name="nome"value="TOB-STT"/>.

Tendo a finalidade de oferecer informações semânticas às categorias as bases de conhecimento do agente foram organizadas por tópicos. Essas categorias são utilizadas quando o sujeito começa a questionar o agente sobre assuntos associados ao tópico. Um exemplo de base de conhecimento que possui um conjunto de categorias estruturadas sobre um determinado assunto é apresentado no Código-fonte 2. Nele é possível perceber que foi utilizado o comando <topic> para definir o nome do tópico. Os pares conversacionais implementados em categorias continuam sendo codificados da mesma forma.

Código-fonte 2 – Exemplo de definição de tópicos de conversação

```
 1: <aiml>
 2:   <topic name="Teste de Mutação">
 3:     <category>
 4:       <pattern> Cite uma ferramenta para automatizar o Teste de
      Mutação </pattern>
 5:         <template>
 6:         Proteum é uma excelente ferramenta para apoiar o teste de
      mutação construído em C. Se o seu software for implementado em
      Java, recomendo a MuJava.
 7:         </template>
 8:     </category>
 9:     <category>
10:       <pattern> Quantos operadores de mutação existem para a
      linguagem C? </pattern>
11:         <template>
```

```
12:         A linguagem C tem 75 operadores.
13:         </template>
14:     </category>
15:   </topic>
16: </aiml>
```

A *tag* <topic> é utilizada para diferenciar o tema da conversação e para casos em que é possível utilizar o mesmo padrão de entrada em contextos diferentes, apresentando saídas diferentes. Uma pergunta pode fazer sentido para um contexto e um sentido para outro. Fazendo uma analogia de uma conversa entre humano, quando uma pessoa está questionando a outra sobre um determinado assunto, ela pode utilizar o mesmo padrão de entrada para se referir a outro assunto.

Quando o tópico da conversa é teste funcional, por exemplo, o aluno pode perguntar ao TOB-STT quais são os critérios da técnica. A mesma pergunta pode ser feita quando o tópico da conversa for teste estrutural. Em AIML é possível fazer uso da *tag* <topic> para diferenciar o contexto e um comando composto pelas tags <think> e <set> para especificar a mudança de tópico durante a conversação. O exemplo apresentado no Código-fonte 3 demonstra o uso desses comandos para construir um cenário de conversação adequado ao assunto da conversa.

Código-fonte 3 – Exemplo de uso do comando <topic>

```
1:  <aiml>
2:    <topic name="Teste Funcional">
3:      <category>
4:        <pattern>Quais são os critérios da técnica?</pattern>
5:          <template>Vou citar os que eu conheço: Particionamento
      em Classes de Equivalência, Análise do Valor Limite, Teste
      Funcional Sistemático, Grafo Causa-Efeito e Error-Guessing. </
      template>
6:        </category>
7:    </topic>
8:    <topic name="Teste Estrutural">
9:      <category>
10:       <pattern>Quais são os critérios da técnica?</pattern>
11:         <template>  O teste estrutural possui diversos critérios
      que são agrupados em categorias, a saber: critérios baseados
      na complexidade, critérios baseados em fluxo de controle, crité
      rios baseados em fluxo de dados e critérios baseados na
      cobertura. </template>
12:     </category>
```

```
13:     </topic>
14:       <category>
15:         <pattern>vamos falar sobre Teste Estrutural</pattern>
16:           <template>Ok.
17:           <think><set name="topic">Teste Estrutural</set></think>
18:         </template>
19:       </category>
20:       <category>
21:         <pattern>vamos falar sobre Teste Funcional</pattern>
22:           <template>  Ok.
23:           <think><set name="topic">Teste Funcional</set></think>
24:           </template>
25:       </category>
26:       <category>
27:         <pattern>sobre o que estamos falando?</pattern>
28:           <template><get name="topic"/></template>
29:       </category>
30:   </aiml>
```

Discursivo

A linguagem AIML oferece suporte ao tratamento de informação no nível discursivo, tendo como base a substituição de pronomes, semelhantes à substituição morfológica. No contexto do conhecimento do TOB-STT, a base de substituição de pronomes é construída com as *tags* <person> e <substitute>. Essa base é utilizada para indicar que um pronome está sendo substituído por outro. Um exemplo é demonstrado: <person> <substitute find="ela é"replace="ele é"/>; </person>.

Pragmático

O nível pragmático diz respeito ao uso de variáveis que armazenam temporariamente informações sobre o sujeito que está interagindo com o TOB-STT. As variáveis são definidas pelo comando <set> e recuperadas por <get>, inseridos nos padrões e moldes de réplica das categorias da base de conhecimento. Informações sobre o usuário podem ser coletadas como o nome do ser humano, dentre outros.

Outros tipos de tratamento

Também foram utilizados comandos que possibilitam simular uma memória na conversação. Em uma conversa entre seres humanos, é comum um sujeito questionar o outro com o objetivo de fazer com que ele seja mais específico sobre um assunto. Por exemplo, em uma conversa entre um aluno e o TOB-STT, o aluno poderia pedir ao agente para ele descrever sobre as técnicas de teste. O agente, então, poderia questioná-lo visando saber exatamente sobre qual a técnica que o aluno deseja receber informações.

Então, o aluno especifica qual a técnica e o agente descrever sobre ela. O Código-fonte 4 apresenta o exemplo implementado em AIML.

Código-fonte 4 – Exemplo de comando para manter o contexto do diálogo

```
1:  <aiml>
2:    <category>
3:      <pattern> * técnicas de teste </pattern>
4:        <template> Mas sobre qual técnica de teste você gostaria
     de saber? </template>
5:    </category>
6:    <category>
7:      <pattern> de caixa preta </pattern>
8:        <that> Mas sobre qual técnica de teste você gostaria de
     saber? </that>
9:          <template> Teste de caixa preta é uma técnica que
     utiliza como fonte de informação a documentação do software e o
      profissional que realiza a atividade de teste não tem acesso
     ao código-fonte do software em teste.
10:         </template>
11:   </category>
12: </aiml>
```

Ao mesmo tempo em que uma pergunta pode levar a diferentes respostas, uma resposta pode ser utilizada para responder diferentes perguntas. Por exemplo, as perguntas "você sabe o que é teste de software?", "você pode definir teste de software?", "me diz o que é teste de software" podem ser respondidas com a mesma resposta "é uma atividade que tem como objetivo revelar a presença de defeitos em um software".

A linguagem AIML foi planejada para possibilitar que o agente conversacional receba diferentes perguntas que tem a mesma intenção e respondê-las com o mesmo padrão de resposta. Wallace (2009) as denominou de estratégia para redução simbólica (do inglês: *symbolic reduction*). Para tanto, é necessário utilizar a *tag* <srai>. Com o intuito de tornar a interação entre aluno e TOB-STT mais agradável, a estratégia para redução simbólica foi adotada. O Código-fonte 5 apresenta um exemplo de uso da *tag*. Assim, se a mensagem enviada pelo usuário estiver em um dos padrões de entrada, ela é encaminhada para um <template> padrão de retorno.

Código-fonte 5 – Exemplo de implementação de redução simbólica

```
1:  <aiml>
2:    <category>
3:      <pattern> você sabe o que é teste de software? </pattern>
```

```
4:        <template> <srai>TSOFTWARE</srai> </template>
5:    </category>
6:    <category>
7:      <pattern> defina teste de software </pattern>
8:        <template> <srai>TSOFTWARE</srai> </template>
9:    </category>
10:   <category>
11:     <pattern> me diz o que é teste de software </pattern>
12:       <template> <srai>TSOFTWARE</srai> </template>
13:   </category>
14:   <category>
15:     <pattern> qual o objetivo do teste de software? </pattern>
16:       <template> <srai>TSOFTWARE</srai> </template>
17:   </category>
18:   <category>
19:     <pattern>TSOFTWARE</pattern>
20:       <template> é uma atividade que tem como objetivo revelar a
      presença de defeitos em um software. </template>
21:     </category>
22: </aiml>
```

Além de redução simbólica, utiliza-se dividir e conquistar. Este método realizada o oposto do método redução simbólica. Ele é usado quando existe um padrão de entrada que pode ter várias saídas possíveis. Desse modo, um padrão de entrada é reconhecido e dividido em sentenças. Cada sentença possui um padrão de saída. No final, o método concatena os padrões de saída possíveis e apresenta a mensagem ao usuário.

O Código-fonte 6 foi construído para exemplificar o uso do método dividir e conquistar. Nele, o usuário faz a seguinte ponderação ao TOB-STT "então particionamento de classes equivalentes é um critério do teste funcional?". Como na base de conhecimento não existe um padrão que contenha exatamente a frase digitada pelo usuário, o interpretador aplica o método dividir e conquistar e utiliza outras entradas. Como resposta o TOB-STT concatena as respostas de padrões apresentados nas linhas 13 e 17.

Código-fonte 6 – Exemplo de implementação do Método dividir e conquistar

```
1:  <aiml>
2:    <category>
3:      <pattern>O que é particionamento de classes de equivalência<
     /pattern>
4:        <template>É um critério da técnica de Teste Funcional que
     divide o domínio de entrada em classes de equivalência (válidas
```

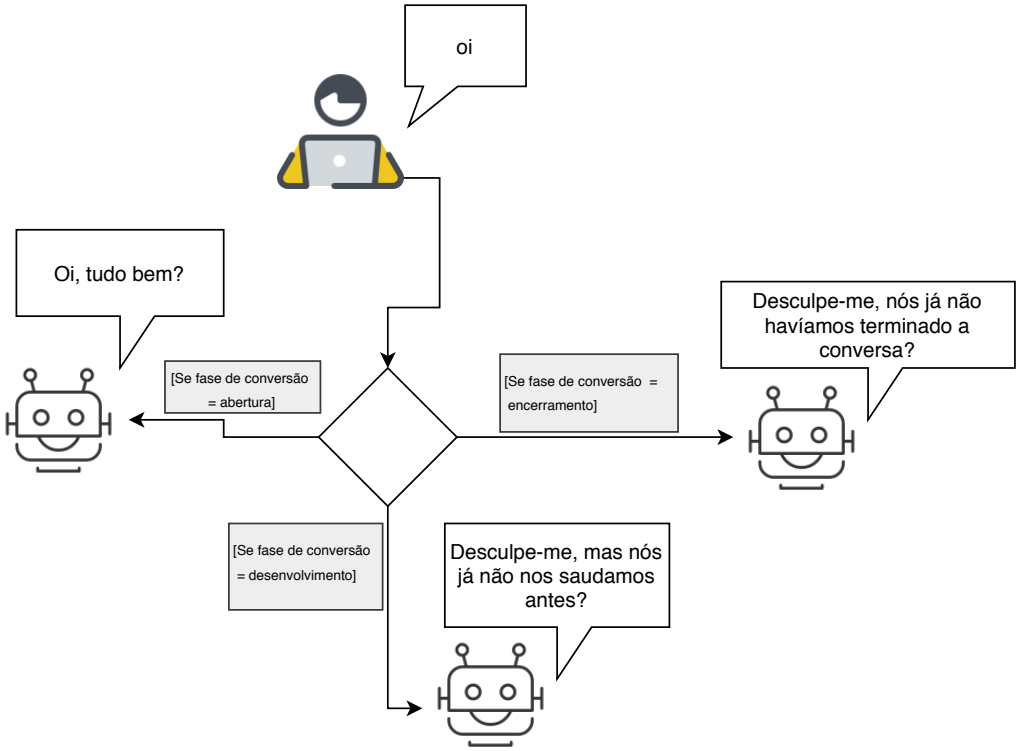
```
      e inválidas), a partir das condições de entrada do programa,
     considerando a especificação do mesmo.
5:        </template>
6:    </category>
7:    <category>
8:      <pattern>Então particionamento de classes equivalentes *</
     pattern>
9:        <template><srai>PCE</srai><sr/></template>
10:   </category>
11:   <category>
12:     <pattern>PCE</pattern>
13:       <template>Sim, o particionamento de classes equivalentes é
      um dos critérios do teste funcional. </template>
14:   </category>
15:   <category>
16:     <pattern>critério do teste funcional</pattern>
17:       <template> A técnica de Teste Funcional tem diversos crité
     rios, tais como: Particionamento em Classes de Equivalencia, An
     álise do Valor Limite, Teste Funcional Sistemático, Grafo Causa
     -Efeito e Error-Guessing. Se você quiser, eu posso especificar
     e mostrar exemplos de cada um desses critérios.
18:       </template>
19:   </category>
20: </aiml>
```

Para tratar as três fases da estrutura da conversação, a linguagem AIML possibilitou o uso de desvios condicionais com variáveis e uma *tag* especifica denominada <condition>. Uma variável é definida com a *tag* <set> e recuperada com o comando <get>, dentro do comando <think>. Visando exemplificar o uso da tag <condition> como apoio para a estrutura de conversação, foi criado um exemplo ilustrado pela Figura 53.

No exemplo, o usuário digita a palavra "oi". Se a conversa estiver na fase de abertura, o TOB-STT responderá com uma mensagem, perguntando como está o usuário, pois é algo comum em uma conversa feita por seres humanos. Se a conversa estiver na fase de desenvolvimento, a mensagem é vista como algo incomum em uma conversa porque é uma mensagem de cumprimentos, então o TOB-STT devolverá uma pergunta dando a impressão de que eles já havia se cumprimentado. Se a conversa estiver na fase de fechamento, a mensagem também é vista como não comum e o TOB-STT responde questionando o usuário sobre o fim da conversa.

Considerando esse exemplo, foi criado um código para demonstrar o uso da estrutura de decisão no desenvolvimento de uma base de conhecimento em AIML. O Código-

Figura 53 – Exemplos de conversa nas três fases de uma conversação



Fonte: Elaborada pelo autor.

fonte 7 demonstra os códigos do exemplo apresentado pela Figura 53. No exemplo que será apresentado, foi implementado desvios condicionais com auxílio da *tag* . É importante salientar que existem diversas formas de implementação dos desvios condicionais, inclusive, segundo (WALLACE, 2009) é possível implementá-los sem uso da *tag* <condition>, fazendo uso da *tag* <srai>.

Código-fonte 7 – Exemplo de implementação de estrutura condicional

```
1:  <aiml>
2:    <!-- FASE DE ABERTURA -->
3:      <category>
4:        <pattern>OI</pattern>
5:          <template>
6:            <condition>
7:              <li name="entrada" value="desenvolvimento">
8:              Desculpe-me, mas nós já não nos saudamos antes?
9:              </li>
10:             <li name="entrada" value="fechamento">
11:             Desculpe-me, nós já não haviamos terminado a
      conversa?
12:             </li>
```

```
13:                <li>
14:                Oi, tudo bem?
15:                <think><set name="entrada">desenvolvimento</set></
      think>
16:                </li>
17:              </condition>
18:            </template>
19:        </category>
20:      <!-- FASE DE FECHAMENTO -->
21:        <category>
22:          <pattern>tchau</pattern>
23:            <template>
24:              <condition>
25:                <li name="término" value="1">
26:                Você já havia me dito tchau. Se você quiser terminar
       a conversa, digite END
27:                </li>
28:                <li>
29:                Se você quiser terminar a conversa, digite END
30:                <think><set name="entrada">fechamento</set></think>
31:                <think><set name="término">1</set></think>
32:                </li>
33:              </condition>
34:            </template>
35:        </category>
36:            <category>
37:              <pattern>END</pattern>
38:                <template>
39:                <think><set name="entrada">0</set></think>
40:                <think><set name="término">0</set></think>
41:                Bye bye . . . estou indo nessa!
42:                </template>
43:        </category>
44:    </aiml>
```

No exemplo ilustrado no Código-fonte 7, quando o usuário entrar com a palavra "oi", o interpretador reconhecerá o padrão de entrada "OI" e verificará as condições definidas no padrão de resposta, especificado nas linhas 7, 10 e 13. A partir disso, testará a primeira condição definida, verificando se a variável "entrada" possuir o valor "desenvolvimento", caso a condição for satisfeita o agente retornará como resposta a frase expressa na linha 8, caso contrário testará a próxima condição.

Na segunda condição, o interpretador verificará se a variável “entrada” possuir o valor “fechamento”, e se ela for satisfeita, o TOB-STT retornará o padrão de resposta definido na linha 11. Caso nenhuma dessas condições forem satisfeitas, o TOB-STT retornará o padrão de resposta definido na linha 14. Ao retornar essa resposta, a variável “entrada” recebe o valor “desenvolvimento”. Assim, se o usuário entrar com a palavra “oi”, a primeira condição será satisfeita e o agente passará a retornar como padrão de resposta o que foi predefinido na linha 8, passando a “entender” que a conversa já havia sido inicializada.

Quando o usuário entrar com a palavra “tchau”, o interpretador irá reconhecer o padrão e testará as condições definidas nas linhas 25 e 29. A primeira condição será satisfeita se a variável “término” possuir o valor “1”. Caso contrário, TOB-STT retornará o padrão de resposta definida na linha 30, reconhecendo que a conversa está na fase de fechamento, visto que, a variável “entrada” passa a receber o valor ”fechamento”. Também terá conhecimento sobre a definição da variável “término” que receberá o valor 1, assim, se o usuário entrar novamente com a sentença “tchau”, conseguirá satisfazer a condição definida na linha 25. Por fim, quando o usuário entrar com a palavra END, recomendado pelo agente para finalizar a conversa, TOB-STT encerrará a conversa e as variáveis “entrada” e “término” também serão finalizadas, fazendo com que a última fase da conversação seja finalizada.

Processamento da Linguagem

Conforme mencionado durante a Seção 6.3.2, a máquina de inferência utilizada pelo TOB-STT é o software Program O (versão 2.6.11). No que concerne a capacidade de interpretação, essa máquina tem duas funcionalidades principais. A primeira consiste na normalização do texto digitado pelo usuário e a segunda consiste na execução do algoritmo Graphmaster para identificar o melhor padrão para a entrada (*i.e.*, texto digitado pelo usuário). Cada uma das funcionalidades será detalhada no decorrer desta seção.

O processo de normalização busca padronizar o texto emitido pelo utilizador do TOB-STT. Em um primeiro momento, as abreviaturas, siglas e símbolos que foram digitadas pelo usuário são substituídas pelas palavras ou locuções originais. Na sequência, a entrada é particionada em um número x de sentenças, em que x é o número de sentenças que correspondem a entrada. Para identificar o número de sentenças, as pontuações do texto são respeitadas.

A terceira etapa do processo de normalização consiste em remover todo tipo de pontuação existente em cada uma das partições realizadas durante a etapa anterior. Por fim, a última etapa do processo consiste em converter todos os caracteres que constituem as sentenças em letras maiúsculas. Essa última etapa é realizada com o propósito de minimizar o número de comparações realizadas pelo algoritmo Graphmaster. Assim, evita-

se a diferenciação entre letras minúsculas e maiúsculas. Nesse contexto, o conteúdo que constitui as bases de conhecimento do TOB-STT, durante o registro no Program O, foi convertido automaticamente pelo software para o formato de letras maiúsculas.

Após a normalização da entrada, o algoritmo Graphmaster é executado. Em síntese, o algoritmo tem a finalidade de identificar o padrão (ou um conjunto de padrões) que correspondem a entrada. Nesse sentido, ele precisará percorrer uma estrutura em formato de árvore, uma vez que a base de conhecimento do agente é organizada na máquina de inferência em uma estrutura de dados conhecida como árvore n-ária (*i.e.* uma árvore cujos nós têm graus menores ou iguais a `n`, onde `n` é o número de filhos de cada nó). A árvore é constituída por cada caractere ou palavra registrada na base de conhecimento do agente. Nessa perspectiva, cada caractere ou palavra é transformada em um nó e a relação entre eles é representa pelos arcos.

Diante da estrutura de dados, para identificar a resposta mais adequada a entrada, o algoritmo Graphmaster percorre e compara cada caractere de cada sentença de entrada com cada um dos nós da estrutura, até encontrar um padrão que corresponde ao que foi informado pelo usuário. Desse modo, primeiramente faz a identificação do nó que corresponde ao primeiro caractere ou palavra informada pelo usuário, posteriormente, seleciona o próximo nó que corresponde com o próximo caractere ou palavra da sentença, repetindo o processo até localizar o padrão que corresponde à entrada.

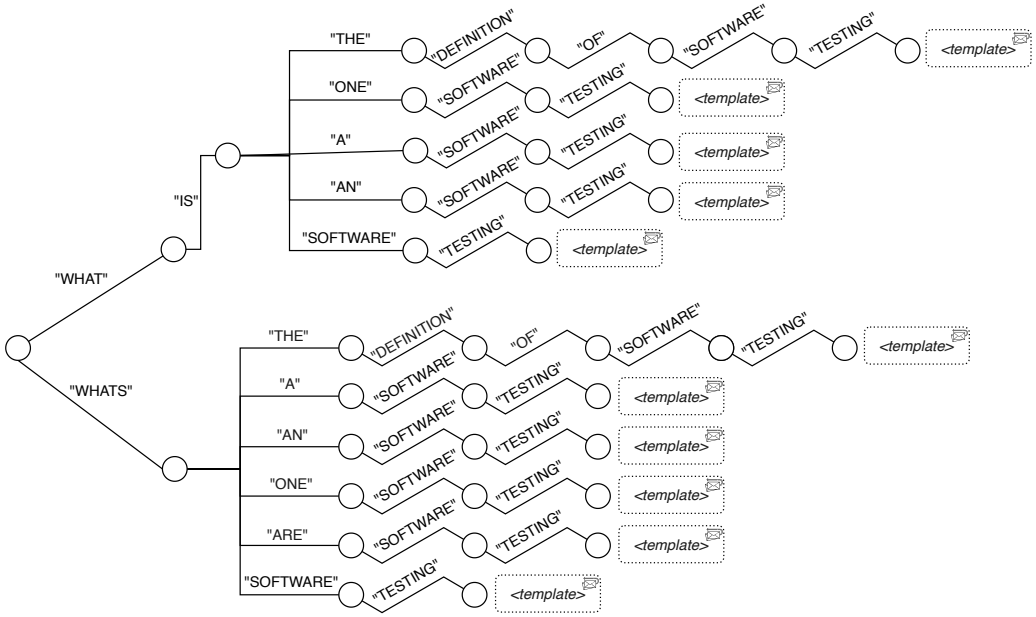
Para ilustrar a estrutura da base de conhecimento do TOB-STT, a Figura 54 foi desenhada. Essa ilustração representa somente onze padrões de entrada que foram construídos para representar o conhecimento do agente. Os padrões estão relacionados a pergunta sobre definição de teste de software. Salienta-se que atualmente, a base de agente é constituída por 83187 padrões. Esses padrões cobrem as três fases de interação, contendo perguntas e respostas sobre cumprimentos, despedidas, terminologias de teste e sobre a técnica de teste funcional e os critérios particionamento em classes de equivalência e análise do valor limite.

6.4 Avaliação do TOB-STT

A primeira avaliação feita levou em conta a qualidade das respostas emitidas pelo agente TOB-STT. Este tipo de avaliação é recomendado por AbuShawar e Atwell (2016), na qual o histórico da interação é analisado.

Para conduzir a avaliação, os parâmetros descritos por AbuShawar e Atwell (2016) foram seguidos. Nesse sentido, o público-alvo do agente conversacional foi delimitado. Na sequência, planejou-se a amostragem, a qual foi definida para ser selecionada por conveniência, caracterizando uma amostragem não probabilística, em que alunos de teste de software, de diferentes graus acadêmicos (graduação e pós-graduação), do Instituto de

Figura 54 – Estrutura que representa uma fração do conhecimento do TOB-STT



Fonte: Elaborada pelo autor.

Ciências Matemáticas e de Computação foram escolhidos para participar.

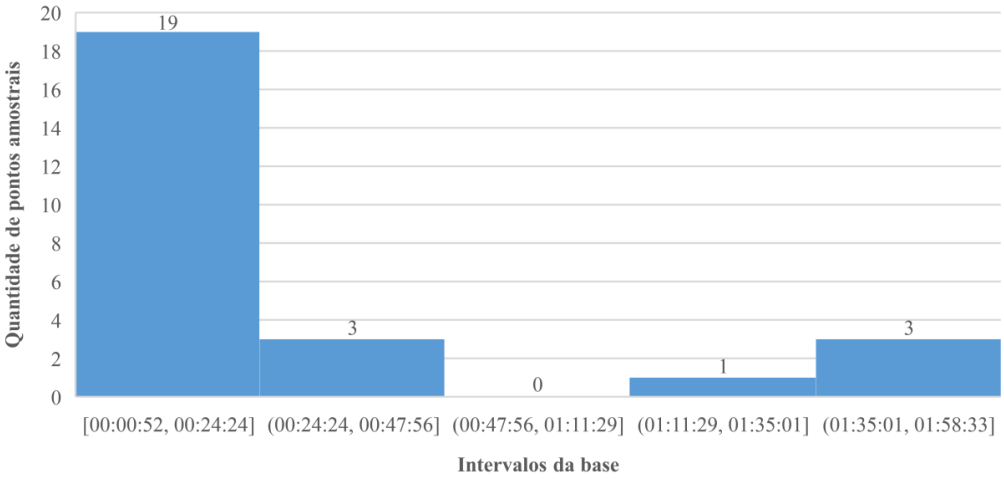
Para apoiar a avaliação, foi preparado um manual de instrução, que contém detalhes sobre o agente e descreve quais são as suas capacidades, incluindo as limitações para resposta. Durante a avaliação, esse material seria recomendado. Um convite para participar do estudo e utilizar o agente TOB-STT foi escrito, em formato de *e-mail*, e encaminhado para estudante de teste de software. No convite, os alunos receberam as instruções de uso, o *link* para acesso ao agente conversacional e foram notificados que deveriam utilizar o agente, observando o manual de uso.

O agente conversacional ficou disponível para ser utilizado pelos participantes durante o período de duas semanas, no início do mês de dezembro de 2018. Após o período, foi feita uma análise nos registros (*logs*) das interações feitas pelos participantes. Foi possível identificar a participação de 26 usuários que interagiram com o agente conversacional. O registro da interação considerou o tempo total que cada usuário interagiu com o agente TOB-STT nesse período.

Uma análise em relação ao tempo de duração das interações foi realizada. Para apoiar a análise, foi considerado a utilização de histograma. Utilizou-se a regra baseada em raiz quadrada para definir o número de partições que constituem o histograma. Como 26 participantes interagiram com o TOB-STT, cinco partições foram geradas. Na Figura 55, é possível observar que a maioria das interações (19, isto é, 73,08%) durou entre 52 segundos a aproximadamente 24 minutos. Três interações (11,54%) ficaram classificadas entre 24 minutos a 47 minutos (aproximadamente). Quatro participantes interagiram com o TOB-

STT por um intervalo de tempo superior a uma hora durante o período de interação.

Figura 55 – Histograma do tempo de duração das interações



Fonte: Elaborada pelo autor.

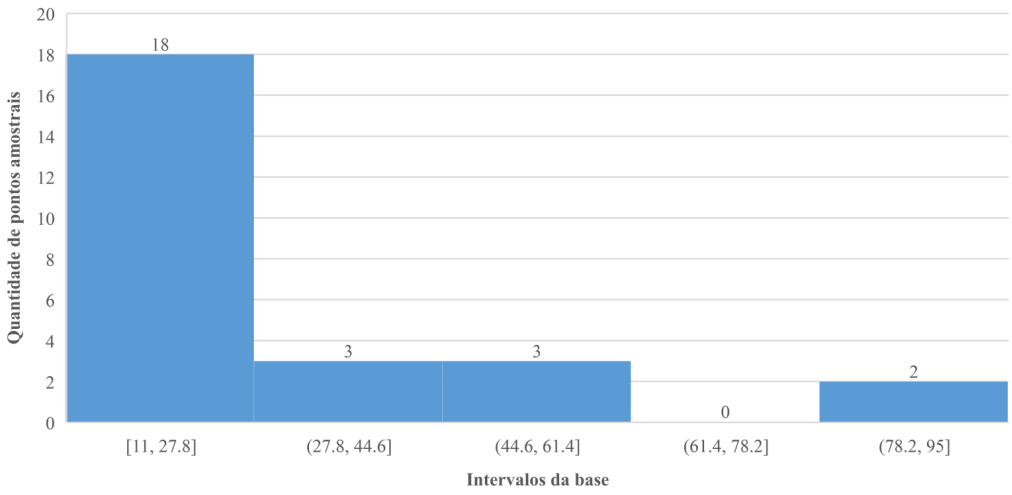
O número de perguntas feitas pelos usuários ao TOB-STT também foi examinado. Similar a análise sobre o tempo de duração das interações, nesta análise também foi utilizado o histograma (Figura 56. A maioria dos participantes (18, isto é, 69,23%) fez entre 11 e 28 perguntas. Três participantes (11,54%) realizaram cerca de 28 e 45 perguntas ao TOB-STT. Outros três indivíduos (11,54%), questionaram perto de 45 e 61 perguntas. Por fim, dois participante (7,69%) fizeram entre 78 e 95 perguntas. Salienta-se que essa análise considera todas as perguntas que foram realizadas por cada usuário, ou seja, perguntas com erros de digitação feitas também foram consideradas.

A Figura 56, demonstra que os participantes, de modo geral, fizeram um número considerável de questionamentos ao TOB-STT, podendo indicar que os sujeitos ficaram interessados com o sistema e/ou com as respostas que ele emite durante a interação. É possível que os participantes que interagiram menos, possam não ter se interessado pelo agente, ou até mesmo não ter conseguindo interagir com ele, em virtude do idioma ou devido a forma de construir as sentenças.

Após a análise geral das perguntas feitas pelos usuários, um filtro foi aplicado, visando eliminar perguntas feitas com erros de digitação, cuja identificação não foi realizada pelo agente conversacional. Além dessas eliminações, também foram excluídas perguntas repetidas (escritas da mesma forma, sem alteração) feitas por um mesmo usuário durante a mesma interação. A partir da eliminação, as perguntas foram analisadas, de acordo com o conteúdo delas e relacionadas com as bases de conhecimento do TOB-STT.

A partir da remoção das entradas consideradas não adequadas, cada pergunta foi classificada de acordo com o seu conteúdo e padrões de entrada (*i.e.*, <pattern>) das bases de conhecimento do agente (Tabela 37). A partir da remoção, restaram-se 451 perguntas

Figura 56 – Histograma do número de perguntas feitas pelos usuários



Fonte: Elaborada pelo autor.

para serem classificadas. A maioria das perguntas realizadas pelos estudantes (58,09%) corresponde aos padrões de entrada das bases de conhecimento que foram realizadas do projeto A.L.I.C.E. Na sequência, a maior quantidade de perguntas está relacionada com conceitos e definições de teste funcional.

Com o intuito de compreender a qualidade das respostas, inspirando-se no estudo de AbuShawar e Atwell (2016), as 451 perguntas foram novamente classificadas, de acordo com a sua qualidade. A classificação considerada o conhecimento do agente e a pergunta feita pelo usuário, a saber:

ideal – a resposta está presente nas bases de conhecimento do agente conversacional e, por consequência, o agente respondeu a pergunta corretamente;

ok – a resposta não está presente nas bases de conhecimento do TOB-STT e, por isso, o agente informou ao usuário que não possui conhecimento sobre o assunto;

problemática – a resposta não está presente nas bases de conhecimento do TOB-STT e o agente apresentou uma resposta incorreta para a pergunta;

erro – a resposta está presente nas bases de conhecimento do agente conversacional, mas o agente não conseguiu localizar a resposta correta.

A Figura 57 contém os resultados da classificação. Foi possível observar que 151 perguntas foram respondidas corretamente pelo agente. Os participantes fizeram 192 perguntas que não estavam previstas e por conta disso foram notificados que o agente não possuía conhecimento sobre o assunto. Por outro lado, notou-se que 108 perguntas foram

Tabela 37 – Classificação dos conteúdos conversados entre TOB-STT e os alunos

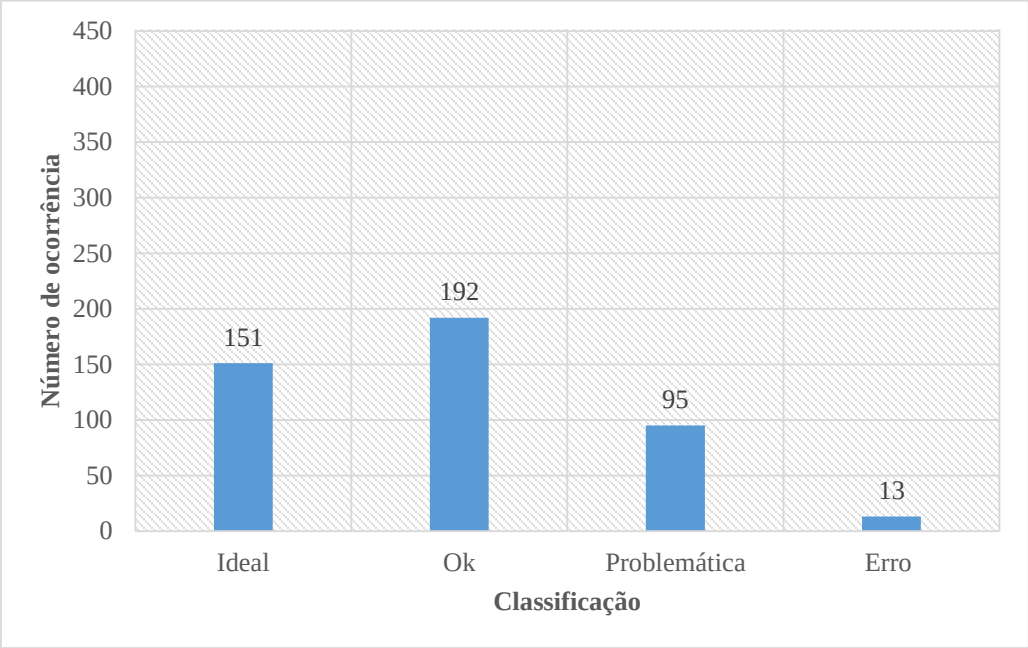
Classificação	Quantidade	Frequência
A.L.I.C.E	262	58,09%
Conceitos de teste funcional	77	17,07%
Terminologia de teste	76	16,85%
Definição, Aplicação e Exemplo de Particionamento em Classes de Equivalência	19	4,21%
Definição, Aplicação e Exemplo de Análise do Valor Limite	16	3,55%
Diferenciação entre critérios de teste	1	0,22%
Total	**451**	100.00%

Fonte: Elaborada pelo autor.

respondidas de forma inadequada pelo agente. Dentre as 108 respostas, 95 deveriam notificar o usuário que o agente não conseguiria responder a pergunta e o usuário deveria reescrevê-la de outra forma e 13 respostas estão relacionadas com perguntas que eram esperadas pelo agente, mas foram respondidas incorretamente pelo agente.

Com base na análise, observou-se que é necessário expandir as bases de conhecimento, uma vez que a avaliação revelou a existência de outros formatos de questões que foram formuladas pelos participantes, cujo as quais não estavam previstas na base de conhecimento do TOB-STT. Também foi possível identificar que as bases reutilizadas do projeto A.L.I.C.E prejudicaram as respostas do agente, uma vez que as 108 perguntas foram respondidas de forma inadequada pelo agente. O TOB-STT, em vez de utilizar as bases sobre o conteúdo de teste de software, respondeu o usuário com as bases de conhecimento do projeto A.L.I.C.E. Com isso, acredita-se que será necessário revisar as bases do projeto A.L.I.C.E, na tentativa de mitigar o problema identificado. Talvez seja necessário definir algum tipo de critério para reutilizar as bases de conhecimento.

Figura 57 – Proporção de cada tipo de resposta identificada



Fonte: Elaborada pelo autor.

Além da análise das interações dos estudantes, outra avaliação do agente foi realizada em conjunto com pesquisadores da área de teste de software (professores, monitores e tutores de disciplinas relacionadas). Os pesquisadores foram selecionados, por conveniência, dos grupos de pesquisa da Universidade de São Paulo e da Universidade Federal de São Carlos.Nessa avaliação, os sujeitos deveriam interagir com o TOB-STT e oferecer um *feedback* sobre a interação.

Para apoiar a avaliação um questionário foi produzido. Esse questionário contém duas perguntas que foram escritas com o propósito de caracterizar a amostragem e outras sete questões que foram escritas com a finalidade de obter as recomendações dos sujeitos e identificar as vantagens e limitações sobre o TOB-STT. As questões que possuem característica avaliativa estão listadas na Tabela 38, algumas foram baseadas nos questionários utilizados durante as avaliações dos agentes Ubibot (PASCHOAL; OLIVEIRA; CHICON, 2018) e ELAI (HERPICH *et al.*, 2016).

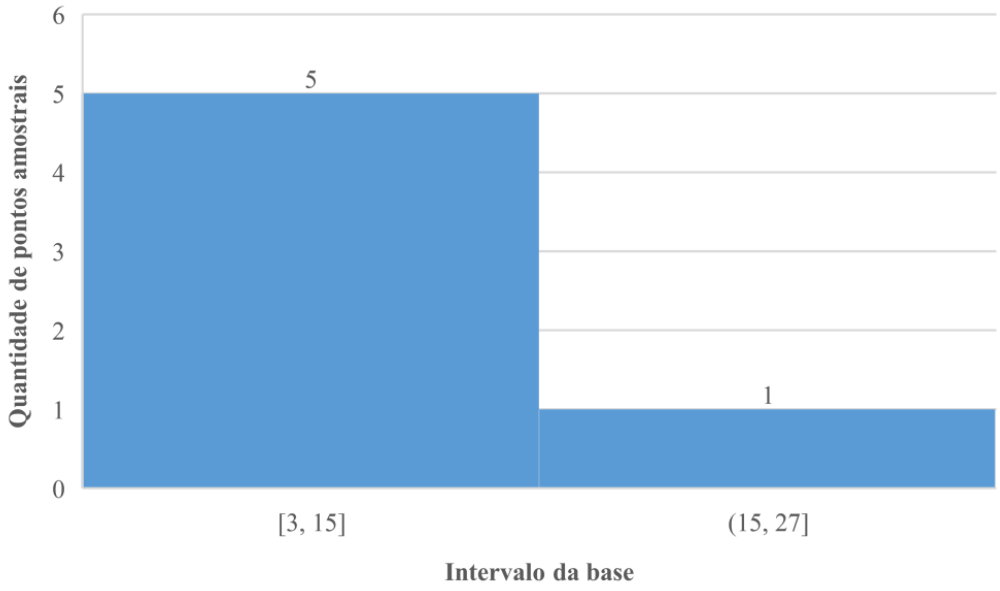
Seis pesquisadores participaram dessa avaliação. Dentre os participantes, 50% informaram que possuíam a titulação de mestre, 33% eram doutores e 17% portavam o título de graduação, com mestrado em andamento. Esse grupo de avaliadores também revelou a experiência (em anos) na área de teste de software. Conforme é possível observar no histograma abaixo (Figura 58), cinco participantes revelaram que possuíam entre 3 e 15 anos de experiência, enquanto um participante mencionou que possui exatamente 27 anos de experiência na área.

Tabela 38 – Questionário produzido para coleta das percepções dos pesquisadores

Questão	Tipo
(1) O TOB-STT fornece respostas claras ao usuário.	Escala Likert (Obrigatória)
(2) O TOB-STT fornece informações confiáveis em suas respostas.	Escala Likert (Obrigatória)
(3) O TOB-STT possui uma interface fácil de usar.	Escala Likert (Obrigatória)
(4) Eu acredito que o TOB-STT poderá ajudar os alunos a solucionarem suas dúvidas sobre o conteúdo de teste de software	Escala Likert (Obrigatória)
(5) Se futuramente eu precisar ministrar um curso sobre teste de software, eu utilizarei o TOB-STT nas minhas aulas para apoiar os alunos.	Escala Likert (Obrigatória)
(6) Use o espaço abaixo para descrever suas observações relacionadas ao uso do TOB-STT.	Aberta (Não obrigatória)
(7) Descreva sugestões de melhoria.	Aberta (Não obrigatória)

Fonte: Elaborada pelo autor.

Figura 58 – Histograma dos anos de experiência dos participantes



Fonte: Elaborada pelo autor.

As cinco questões obrigatórias, continham cinco opções de resposta, as quais variavam de concordo totalmente à discordo totalmente, tendo como base os preceitos da Escala Likert. As respostas obtidas nessas questões estão disponíveis na (Tabela 39). Adicionalmente, tendo o intuito de averiguar a confiabilidade do questionário que foi construído para essa avaliação e, consequentemente, ampliar a credibilidade da avaliação, utilizou-se o Alfa de Cronbach (CRONBACH, 1951).

A partir do cálculo, o valor de $alpha$ para o Alpha de Cronbach foi de $\alpha = 0{,}66$.

Para interpretação do coeficiente alfa de Cronbach, utilizou-se o estudo de Landis e Koch (1977). Diante disso, a consistência interna do questionário pode ser considerada aceitável, uma vez que ficou dentre os valores que compõem a categoria substancial. Salienta-se que o número pequeno de perguntas e respondentes podem ter afetado o valor de $alpha$, uma vez que o número reduzido de itens e participantes tendem a diminuir o seu valor (LANDIS; KOCH, 1977).

Tabela 39 – Respostas dos pesquisadores nas cinco perguntas

	Discordo totalmente		**Discordo Parcialmente**		**Indeciso**		**Concordo Parcialmente**		**Concordo Totalmente**	
	R	**%**	**R**	**%**	**R**	**%**	**R**	**%**	**R**	**%**
Questão 1	0	0,00	2	33,33	2	33,33	2	33,33	0	0,00
Questão 2	0	0,00	1	16,67	0	0,00	1	16,67	4	66,67
Questão 3	1	16,67	0	0,00	1	16,67	2	33,33	2	33,33
Questão 4	0	0,00	0	0,00	3	50,00	2	33,33	1	16,67
Questão 5	2	33,33	1	16,67	2	33,33	0	0,00	1	16,67

Fonte: Elaborada pelo autor.

A análise da primeira questão revela que não houve uma predisposição dos sujeitos em escolher um posicionamento. Enquanto dois participantes discordaram parcialmente, dois concordaram parcialmente. Além disso, dois participantes revelaram que estavam indecisos. Em razão das respostas discordantes e devido ao grupo de participante ter um tamanho reduzido, não foi possível tornar evidente se as respostas do TOB-STT são claras. Durante a segunda questão, cinco participantes concordaram que o TOB-STT fornece informações confiáveis em suas respostas.

Uma questão sobre a facilidade de uso da interface também foi disponibilizada no questionário. Na mesma, os participantes deveriam indicar se concordam que a interface do TOB-STT é fácil de usar (Questão 3). Como resposta quatro participantes concordaram. Um participante discordou totalmente e um participante ficou indeciso. Em relação a questão 4, os participantes assinalaram opções de posicionamento que variam de indecisão (3 respostas) a concordância (3 respostas). Desse modo, nenhum participante acredita que o TOB-STT não conseguirá ajudar os alunos a solucionarem as dúvidas sobre o conteúdo de teste de software.

A questão 5 buscou colocar os participantes em uma situação que deveriam decidir se utilizariam o agente em suas aulas, como mecanismo de apoio ao ensino. Similarmente à Questão 1, houveram diferentes tipos de resposta. Três participantes revelaram que não pretendem. Dois pesquisadores declararam indecisão. Apenas um participante concordou plenamente que utilizará a agente em suas disciplinas.

Na sequência, os pesquisadores apresentaram algumas observações relacionadas ao uso do TOB-STT. As respostas, de um modo geral, foram construtivas. Nota-se que

houve uma preocupação dos participantes em mencionar sobre a necessidade de aumentar o vocabulário do agente, expandir as bases de conhecimento, visando melhorar a comunicação entre o aluno e o agente conversacional. Um dos participantes inclusive alertou que as tentativas de interação feitas pelo agente podem prejudicar o desenvolvimento da conversa. Uma síntese das respostas é apresentada na Tabela 40.

Tabela 40 – Síntese das observações realizadas pelos pesquisadores

Feedback dos professores	**Número de ocorrência**	**Frequência**
Encontrei dificuldade ao usar o agente. É preciso expandir o vocabulário para que haja uma melhor interação do usuário.	2	33,33%
O estágio do desenvolvimento ainda é limitado. Por exemplo, não é possível saber sobre o que o agente pode ou não responder.	1	16,67%
Consegui interagir com o agente, seguindo as recomendações para uso. Particularmente, gostei bastante do agente	1	16,67%
Analisando o agente, eu acho que algumas respostas possam causar um desconforto em alguns usuários, uma vez que em momentos o agente é irônico. Esse e outros tipos de tentativas de interação podem causar um desvio de atenção em alguns usuários	1	16,67%
Eu acho que o agente tem potencial para ajudar, porém ajustes precisam ser realizados para que o agente consiga melhor entender e formar uma resposta	1	16,67%

Fonte: Elaborada pelo autor.

A questão 6 do questionário foi dedicada a sugestões de melhorias. Com base nas recomendações, notou-se a preocupação dos participantes em apresentar orientações para ajustar as bases de conhecimento do TOB-STT e a interface (Tabela 41). Em especial, um avaliador comentou sobre a linguagem que é reconhecida pelo agente. Um avaliador que tem obtido experiências de ensino em outras instituições, comentou que não adotaria a ferramenta em suas aulas, porque nem todos os estudantes tem fluência no idioma inglês e talvez a ferramenta se tornaria mais acessível se entendesse a língua portuguesa.

Tabela 41 – Síntese das sugestões de melhorias apontadas pelos pesquisadores

Sugestões de melhor	**Número de ocorrência**	**Frequência**
Expandir o vocabulário, integrá-lo em redes sociais, fazer uma revisão de inglês em algumas respostas. Acho que a adoção se tornaria mais favorável em redes socais.	2	33,33%
Eu acho que a interface do TOB-STT pode ser melhorada, com recursos de animação durante a resposta. O próprio agente poderia ser representado por um avatar/personagem. Eu também acho que seja interessante expandir as variações de respostas que o agente emite para uma mesma pergunta.	1	16,67%
Acredito que os comentários na questão anterior oferecem uma noção de algumas melhorias que precisam ser feitas. Acho que é importante demonstrar aos alunos que esse tipo de recurso é melhor que uma busca em um arquivo PDF, por exemplo.	1	16,67%
Eu tenho três recomendações principais. (1) o agente apresentou somente um exemplo sobre Particionamento em Classes de Equivalência, mas três sobre Análise do Valor Limite. Acho que poderia expandir a quantidade de exemplos. (2) Para quem é da área, a interação com o agente parece ser trivial, mas para quem não é da área pode não ser. Assim, recomendo que as respostas do agente sejam "menos formais". (3) Linguagem. Já ministrei minicursos com o material (slides) em inglês e praticamente todos os alunos criticaram. Em São Carlos, talvez, seja uma realidade diferente onde os alunos sabem da importância do domínio do inglês e têm oportunidades gratuitas para estudarem, porém em universidades presentes em cidades menores há um cenário diferente. Tenho feito meus experimentos (também sobre o ensino de teste de software) na UFES (campus Alegre) e no IFES (campi Cachoeiro e Alegre), e estas são duas instituições cujos alunos têm pouquíssimas oportunidades de pesquisa, aprendizado do inglês, estágios, etc. A ferramenta desenvolvida é de uma ideia excelente, mas talvez este cenário justificar a importância de a termos também em nossa língua nativa. Por conta disso, eu respondi que não utilizarei a ferramenta nas minhas aulas.	1	16,67%
Não tenho sugestões técnicas, somente um comentário. Eu acho que o agente pode ajudar os alunos durante a resolução de exercícios durante as aulas.	1	16,67%

Fonte: Elaborada pelo autor.

A partir dos resultados, foi possível constatar que as bases de conhecimento do agente ainda precisam ser melhoradas para disponibilização e uso. O projeto de iniciação científica (FAPESP Processo 18/16882-6) deverá abordar as manutenções e melhorias nas bases, levando em consideração as sugestões dos avaliadores. Apesar das melhorias necessárias, em geral, os avaliadores gostaram das respostas emitidas pelo agente, uma vez que declararam que elas são confiáveis. A interface, conforme já mencionado, deverá ser melhor desenvolvida. Contudo, para uma melhoria na interface, serão necessários alguns estudos.

Ao passo que foram feitas avaliações preliminares sobre o TOB-STT, ele ficou

disponível para utilização. Assim, foi possível experimentá-lo como mecanismo de apoio ao *flipped classroom* durante o ensino de teste de software, mas especificamente teste funcional. No decorrer da próxima seção, um estudo sobre a avaliação do agente enquanto mecanismo de apoio ao *flipped classroom* será apresentado.

6.5 TOB-STT: uma experiência de uso

Em busca de subsídios para compreender se o agente conversacional pode contribuir com o aprendizado dos alunos "antes da aula", em uma implementação de *flipped classroom*, uma experiência foi realizada. A experiência consistiu na implantação da abordagem FlipClass-STT, durante o ensino da técnica de teste funcional e o uso do agente conversacional TOB-STT pelos estudantes da disciplina. Assim, ao final de uma aula tradicional, em que o conteúdo de teste de software estava sendo ministrado, os recursos educacionais que constituem a abordagem FlipClass-STT foram disponibilizados para 12 estudantes.

O grupo de alunos que estava frequentando a disciplina foi notificado que deveria estudar preliminarmente o conteúdo dos recursos e utilizar o agente conversacional para tentar solucionar as dúvidas que surgissem durante o momento de estudo. Após a utilização do agente conversacional, os estudantes deveriam responder um questionário, com o propósito de indicar se a interação foi importante para o momento de estudo.

Antes dessa aula, uma avaliação a respeito do agente conversacional foi planejada. A experiência teve a intenção de coletar as percepções dos estudantes em relação ao uso do agente TOB-STT durante o aprendizado do conteúdo de teste funcional com *flipped classroom*, dando atenção ao momento "antes da aula".

Para identificar as percepções sobre o uso do agente conversacional durante o aprendizado com *flipped classroom*, um questionário foi desenvolvido. Esse questionário é composto por dez questões objetivas e uma questão dissertativa, todas obrigatórias (Tabela 42). As questões objetivas são assertivas, baseadas na escala Likert de 5 pontos, que contém opções de respostas entre discordo totalmente até concordo totalmente. Essas afirmativas, colocavam o usuário em cenários que eles teriam que ponderar sobre a capacidade do agente ajudá-los durante o aprendizado, "antes da aula". A questão dissertativa, por sua vez, foi definida para apoiar a última questão objetiva. Nela os alunos deveriam indicar conteúdos que eles acreditam que o agente conversacional seria útil para ajudar os alunos.

Tabela 42 – Questionário para coletar as percepções dos estudantes sobre o agente conversacional como mecanismo de apoio ao momento "antes da aula"

Questão	**Tipo**
(1) Eu fiquei com algumas dúvidas durante o aprendizado de Teste Funcional e seus respectivos critérios (Particionamento em Classes de Equivalência e Análise do Valor Limite)	Escala Likert (Obrigatória)
(2) Eu usei o TOB-STT para solucionar as minhas dúvidas	Escala Likert (Obrigatória)
(3) O TOB-STT solucionou todas as minhas dúvidas	Escala Likert (Obrigatória)
(4) Eu não consegui me comunicar com o TOB-STT	Escala Likert (Obrigatória)
(5) Eu acredito que o TOB-STT poderia me ajudar mais durante o aprendizado de Terminologias e Conceitos básicos sobre teste	Escala Likert (Obrigatória)
(6) Eu acredito que o TOB-STT poderia me ajudar mais durante o aprendizado dos critérios baseados em Fluxo de Controle	Escala Likert (Obrigatória)
(7) Eu acredito que o TOB-STT poderia me ajudar mais durante o aprendizado dos critérios baseados em Fluxo de Dados	Escala Likert (Obrigatória)
(8) Eu acredito que o TOB-STT poderia me ajudar mais durante o aprendizado de Teste de Mutação	Escala Likert (Obrigatória)
(9) Eu acredito que o TOB-STT poderia me ajudar mais durante o manuseio de ferramentas de automatização de teste	Escala Likert (Obrigatória)
(10) Eu acredito que o TOB-STT poderia me ajudar a aprender outros conteúdos sobre Teste de Software	Escala Likert (Obrigatória
(11) Indique os conteúdos	Aberta (Não obrigatória)

Fonte: Elaborada pelo autor.

Nove alunos responderam ao questionário. As respostas obtidas nas dez questões foram analisadas e estão representadas na Tabela 43. Além disso, com a finalidade de verificar a confiabilidade do questionário que foi construído para essa avaliação e, consequentemente, ampliar a credibilidade da avaliação, utilizou-se o Alfa de Cronbach (CRONBACH, 1951).

Tabela 43 – Percepções dos estudantes sobre o agente conversacional durante o momento "antes da aula"

	Discordo totalmente		**Discordo Parcialmente**		**Indeciso**		**Concordo Parcialmente**		**Concordo Totalmente**	
	R	**%**	**R**	**%**	**R**	**%**	**R**	**%**	**R**	**%**
Questão 1	3	33,33	2	22,22	3	33,33	1	11,11	0	0,00
Questão 2	3	33,33	2	22,22	2	22,22	0	0,00	2	33,33
Questão 3	6	66,67	2	22,22	0	0,00	0	0,00	1	16,67
Questão 4	3	33,33	2	22,22	1	11,11	1	11,11	2	33,33
Questão 5	2	22,22	2	22,22	1	11,11	1	11,11	3	50,00
Questão 6	2	22,22	1	11,11	2	22,22	1	11,11	3	50,00
Questão 7	2	22,22	1	11,11	2	22,22	1	11,11	3	50,00
Questão 8	2	22,22	1	11,11	2	22,22	2	22,22	2	33,33
Questão 9	2	22,22	1	11,11	1	11,11	1	11,11	4	66,67
Questão 10	2	22,22	1	11,11	2	22,22	0	0,00	4	66,67

Fonte: Elaborada pelo autor.

A partir do cálculo, o valor de *alpha* para o Alpha de Cronbach foi de $\alpha = 0{,}89$. Para interpretação do coeficiente alfa de Cronbach, utilizou-se o estudo de Landis e Koch (1977). Diante disso, a consistência interna do questionário pode ser considerada aceitável, uma vez que ficou dentre os valores que compõem a categoria "quase perfeito".

Em relação à primeira questão, os estudantes tiveram que indicar se ficaram com dúvidas no decorrer do aprendizado de teste funcional a partir dos recursos educacionais da abordagem FlipClass-STT. Somente um aluno indicou que havia ficado com dúvida. As demais respostas variam entre indeciso (3 respostas) e discordo (3 para discordo totalmente e 2 para discordo). Esse resultado pode indicar que o agente conversacional pode não ser tão útil durante o aprendizado desse conteúdo, uma vez que um aluno mencionou que havia ficado com algum tipo de dúvida.

Na segunda questão, os participantes tiveram que indicar se eles utilizaram o agente conversacional para solucionar as suas dúvidas. Apesar da resposta da questão anterior ter indicado que somente um aluno ficou com dúvida, nesta questão, dois alunos indicaram que utilizaram o agente para solucionar as suas dúvidas.

Quanto à terceira questão os alunos tiveram que revelar se o agente conversacional solucionou todas suas dúvidas. O aluno indicou que na primeira questão havia ficado com dúvidas durante o estudo com o conteúdo da abordagem FlipClass-STT, nesse momento indicou que o TOB-STT solucionou as suas dúvidas. Apesar disso, os demais respondentes discordaram. Acredita-se que a não concordância pode ter relação com o fato de os alunos não terem ficado com dúvidas. Também é possível que as respostas obtidas possam ter relação com o fato da necessidade de expansão das bases de conhecimento.

Durante a quarta questão, os estudantes indicaram se haviam conseguido se comunicar com o agente TOB-STT. Cinco participantes declararam que não conseguiram se comunicar direito com o agente, enquanto dois conseguiram interagir de maneira adequada. Um aluno declarou que estava indeciso quanto a sua resposta. Os dados indicam que a maioria dos alunos teve dificuldade de se comunicar com o agente. É possível que esses indivíduos tenham tido dificuldade por não seguir as orientações para uso, uma vez que um grupo menor de estudantes da disciplina conseguiu interagir. É provável também que os alunos não tenham um bom domínio na redação de texto/perguntas no idioma inglês.

A partir da quarta questão, os estudantes teriam que analisar alguns tópicos da área de teste de software e indicar se acreditavam que o TOB-STT poderia ajudá-los durante o estudo dessas temáticas. Na quinta questão, a análise deveria ser realizada acerca da temática de Terminologias e Conceitos básicos sobre teste. Para esse tópico, quatro outros alunos indicaram que concordavam. Um aluno mencionou indecisão e restante acredita que o agente não poderia ajudar quando o aluno estivesse estudando o presente tópico.

Em relação à sexta questão, os estudantes tiveram que considerar o aprendizado dos critérios baseados em fluxo de controle. Para essa questão, três alunos revelaram que acreditam que o agente não pode ajudar os alunos durante o aprendizado desse conteúdo. Dois alunos indicaram indecisão e a maioria dos estudantes revelou que acredita que o agente pode ajudar durante o aprendizado desse tópico.

Os critérios baseados em Fluxo de dados também foram analisados pelos alunos. Para esse conteúdo, a maioria dos alunos indicou que acredita que o TOB-STT poderia ajudá-los durante o aprendizado. Apesar disso, três alunos discordaram e dois alunos ficaram indecisos. Sobre o aprendizado de teste de mutação, quatro alunos indicaram que acreditam que o agente pode ajudar durante o aprendizado desse critério. Por outro lado, três alunos discordam e dois alunos prefeririam não opinar.

Durante a nona questão, os alunos tiveram que opinar se eles acreditavam que o agente poderia ajudá-los durante o manuseio de ferramentas de teste. Para essa questão, cinco alunos concordaram, um aluno preferiu não indicar um posicionamento e outros três alunos indicaram que discordavam.

Durante a última questão objetiva, os alunos deveriam refletir e indicar se eles acreditavam que o TOB-STT poderia ajudá-lo a aprender outros conteúdos sobre teste de software. Com base nela, quatro alunos indicaram que concordavam, dois alunos mencionaram que estavam indecisos e três alunos discordaram.

A partir dessa resposta, os alunos deveriam indicar quais os conteúdos que eles achavam que o agente poderia ser mais útil para apoiar o aprendizado. Apenas dois alunos indicaram os conteúdos, os demais não souberam indicar. Dentre as indicações, um aluno mencionou que o agente seria útil durante o apoio do uso de ferramentas de teste. Outro aluno indiciou que o agente poderia ajudá-los durante o desenvolvimento de casos de teste. As indicações dos estudantes são interessantes e podem ser consideradas durante a evolução do TOB-STT. As bases de conhecimento não foram planejadas para apoiar o uso de ferramentas de teste, mas a sugestão do aluno é algo que deve ser revista. A outra indicação também é bastante relevante, uma vez que as bases de conhecimento atuais foram planejadas para apoiar o aluno a compreender sobre como identificar os requisitos de teste, mas não a construir os casos de teste.

Com base nos resultados das avaliações, acredita-se que foi possível obter uma visão geral sobre as possibilidades de uso do TOB-STT como mecanismo de apoio ao *flipped classroom*. Os dados indicam que nem todos utilizaram o agente para solucionar as dúvidas sobre o conteúdo que estavam estudando. Mesmo assim, alguns alunos indicaram que acreditam que o agente pode ser útil para o aprendizado de outros conteúdos.

Quadro 7 – Conteúdo indicados pelos estudantes

	Resposta
1	Conteúdos teóricos, como conceitos básicos do teste de software, pode servir também como apoio no uso de ferramentas como um instrutivo ativo.
2	Desenvolvimento de casos de teste no uso de algum critério
3	Nenhuma indicação
4	Nenhuma indicação
5	Como realizar os testes e o que são cada tipo de teste.
6	Nenhuma indicação
7	Acho que pode ajudar no aprendizado de qualquer conteúdo.
8	Nenhuma indicação
9	Nenhuma indicação

Fonte: Elaborada pelo autor.

6.6 Considerações Finais

Este capítulo teve o propósito de investigar o agente conversacional como mecanismo de apoio ao modelo pedagógico *flipped classroom*. Para tanto, um agente conversacional foi desenvolvido com o propósito de apoiar o aprendizado de teste de software. Esse agente intitulado TOB-STT é capaz de apresentar dicas e solucionar dúvidas dos estudantes, desde que essas sejam sobre conceitos, aplicações de critérios e exemplos. Após o desenvolvimento de algumas bases de conhecimento, em especial das relacionadas ao conteúdo de teste funcional, o agente conversacional foi avaliado com estudantes e pesquisadores da área de teste de software. Após essas avaliações, o agente foi disponibilizado para alunos que estavam estudando teste funcional com o modelo *flipped classroom*. Com base nisso, foi possível observar que alguns alunos não utilizaram o agente para solucionar as dúvidas sobre o conteúdo, mas acreditam que ele pode ser útil para solucionar as dúvidas sobre outros temas.

Ressalta-se que o resultado da utilização do TOB-STT pelos alunos pode ter haver com a ordem que o conteúdo foi ministrado. Antes do agente conversacional ser disponibilizado aos alunos, os estudantes já haviam tido contato com outros conteúdos na disciplina de teste de software (*e.g.*, critérios baseados em fluxo de controle, critérios baseados em fluxo de dados, teste de mutação), por meio do modelo *flipped classroom*. Como os alunos já tinham visto os conteúdos e por teste funcional ser uma técnica mais simples, os alunos podem ter respondido que o TOB-STT não era tão adequado a essa técnica. É possível que os resultados fossem diferentes se eles não tivessem estudado preliminarmente os outros conteúdos.

CAPÍTULO

7

CONCLUSÕES

7.1 Considerações Iniciais

Este capítulo tem a finalidade de apresentar os principais resultados obtidos com a realização deste projeto de mestrado. Assim, as principais contribuições científicas, tecnológicas e de formação de recursos humanos abrangidas por este trabalho são apresentadas. Adicionalmente, este capítulo é dedicado a apresentação das limitações desta pesquisa, as quais devem ser levadas em consideração durante a interpretação dos resultados obtidos. As produções científicas geradas no período de desenvolvimento desta pesquisa também são apresentadas. Por fim, com o intuito de estimular a continuidade do trabalho, são apresentadas sugestões para o desdobramento desta pesquisa.

7.2 Principais descobertas

Durante o desenvolvimento deste projeto, buscou-se estudar e oferecer direções para três problemas de pesquisa da área: (i) É viável ensinar conteúdos de teste de software com o modelo pedagógico *flipped classroom*? ; (ii) Como implementar o modelo *Flipped Classsroom* no ensino de teste de software? ; e, (iii) Um agente conversacional é útil para o modelo *flipped classroom* no contexto de uma aula de teste de software?. Procedimentos foram realizados na busca por respostas das questões.

Em relação à primeira questão, três estudos experimentais foram realizados, observando o ensino da técnica de teste funcional. A partir dos achados, foi possível reconhecer que o modelo *flipped classroom* produz resultados positivos no ensino desse conteúdo, uma vez que promove uma melhoria no aprendizado dos alunos, possibilitando que o aluno que estuda com esse modelo tenha uma efetividade maior durante a identificação de requisitos de teste que os alunos que estudam com o modelo tradicional. Apesar dessas descobertas,

o tempo que o aluno "gasta" para estudar "antes da aula" é maior que no modelo tradicional de ensino, o desempenho do aluno é o mesmo que no modelo tradicional e o modelo não afeta a motivação do aluno.

Quanto à segunda questão de pesquisa, em razão da experiência adquirida por meio dos estudos experimentais e da revisão de literatura, foi possível identificar as atividades necessárias para a implantação do modelo *flipped classroom* no ensino de teste de software. Essas atividades envolvem a definição do conteúdo, o estabelecimento dos objetivos de aprendizagem para o conteúdo, identificação e seleção de recursos educacionais os quais serão usados pelos alunos, escolha de atividades cujo propósito sejam avaliar se os objetivos de aprendizagem foram atingidos. Buscando facilitar a implementação do modelo, a abordagem FlipClass-STT foi definida em conjunto com uma ferramenta que busca mantê-la sustentável.

Por último, foi necessário identificar as estratégias adequadas para a implementação de um agente conversacional. A partir de um mapeamento sistemático, foi possível revelar que os agentes conversacionais para apoio ao ensino vêm sendo produzidos com a técnica de Casamento de Padrões e a Linguagem AIML. Um agente conversacional, denominado TOB-STT, foi desenvolvido em conjunto com um projeto de iniciação científica. Foi necessário avaliar o agente em relação à qualidade das respostas emitidas, sobre a percepção dos pesquisadores e finalmente, foi possível obter as percepções de estudantes de teste de software sobre o uso do agente no momento "antes da aula", durante a implementação de *flipped classroom*. Os resultados da aplicação do agente com a turma de alunos revelam que o agente não foi proveitoso para o conteúdo de teste funcional, mas os estudantes acreditam que ele pode ser proveitoso para outros assuntos como o teste estrutural, teste de mutação, ferramentas de teste e durante a construção do conjunto de casos de teste.

7.3 Contribuições

Dentre as principais contribuições deste trabalho pode-se citar:

- Avaliação da aplicabilidade do modelo *flipped classroom* no ensino de teste de software;
- Disponibilização de três pacotes de laboratório que podem ser utilizados em novos estudos experimentais sobre ensino de teste de software funcional, os quais podem facilitar a comparação dos resultados obtidos com outros modelos pedagógicos e até mesmo outras técnicas de teste;
- Evidências sobre o estado da prática de ensino de teste de software no Brasil, especialmente relacionado aos conteúdos que são ensinados, métodos de ensino, meca-

nismos de apoio adotados pelos professores e estratégias utilizadas para avaliar os estudantes;

- Definição de abordagem de apoio à implementação do modelo *flipped classroom* no ensino de teste de software (FlipClass-STT), que abrange os principais assuntos referentes a teste de software, de acordo com a pesquisa realizada neste trabalho;

- Aplicação da abordagem FlipClass-STT no ensino de teste de software e aprendizado adquirido por meio da implementação, com direcionamentos para novas aplicações;

- Estabelecimento de um mecanismo de apoio à sustentabilidade da abordagem FlipClass-STT;

- Disponibilização do material produzido para acompanhar o aprendizado dos estudantes durante a implementação da abordagem FlipClass-STT, os quais poderão ser utilizados para ensinar teste de software, na produção de recursos educacionais abertos e até mesmo em estudos experimentais e/ou estudos empíricos relacionados com a área;

- Identificação do estado da arte referente ao processo de desenvolvimento de agentes conversacionais para educação;

- Concepção de um agente conversacional para apoiar o aprendizado de teste de software (TOB-STT), capaz de reconhecer as dúvidas dos alunos por meio de texto no idioma inglês, podendo atuar como um companheiro virtual de aprendizado "antes da aula";

- Formação de recursos humanos na área de teste de software, por meio do estabelecimento projeto de iniciação científica "Chatterbot: uma contribuição ao ensino de teste de software", que recebe apoio da FAPESP;

- Direcionamentos sobre a repercussão de um agente conversacional para solucionar as dúvidas dos estudantes durante o momento "antes da aula", em implementações do modelo *flipped classroom*;

- Subsídios para o desdobramento de novos projetos de pesquisa nas temáticas de ensino de teste de software, *flipped classroom* e agentes/sistemas conversacionais.

7.4 Limitações

O presente estudo possui algumas limitações, descritas a seguir:

- Os estudos de viabilidade conduzidos abordaram apenas a técnica de teste funcional. Por conta disso, é possível que os mesmos efeitos produzidos pelo modelo *flipped classroom* não sejam obtidos em outras técnicas e critérios de teste de software.

- A abordagem FlipClass-STT está condicionada aos assuntos que são ensinados por professores que responderam ao *survey*. Com base nisso, é possível que os assuntos que são contemplados por ela não sejam ensinados pela maior parte dos professores da área, somente pelos respondentes. Além disso, a abordagem FlipClass-STT é resultante de experiência adquirida com base em implementações do modelo e de estudos feitos em artigos científicos. Não foi possível adquirir a experiência com profissionais que possuem vivência na área e utilizam o modelo *flipped classroom*.

- O agente conversacional está limitado ao idioma inglês, compreende padrões pré-estabelecidos e foi desenvolvido em uma linguagem reconhecida por meio do mapeamento sistemático. O mapeamento, por sua vez, considerou bases de dados científicas reconhecidas da área de Ciência da Computação, mas não todas bases de informações disponíveis, como relatórios técnicos e teses e dissertações da área, o que pode limitar a identificação. As bases de conhecimento, até o presente momento estão limitadas ao conteúdo de terminologia e teste funcional.

7.5 Trabalhos Futuros

Com base nas contribuições deste trabalho, alguns estudos futuros podem ser conduzidos:

- Investigar a viabilidade de ensinar técnicas de teste estrutural e teste de mutação com o modelo *flipped classroom*;

- Estudar as ferramentas de apoio à automatização de teste de software que são ensinadas durante o ensino de teste de software. Com base nisso, estipular mecanismos de apoio ao ensino dessas ferramentas, considerando os objetivos de aprendizagem definidos ao longo da abordagem, buscando expandir o número de ferramentas de apoio que foram definidas na FlipClass-STT;

- Investigar problemas reais da indústria do software e transcrevê-los em atividades práticas para serem utilizadas como exercícios no ensino de teste de software, visando estimular a utilização de problemas do "mundo real" no ensino de teste de software, mais especificamente no momento "durante a aula";

- Estender a ferramenta Catolog-STT para que ela possibilite oferecer os recursos educacionais em formato de serviço (web service). Assim, ambientes virtuais de

aprendizagem e outros sistemas educacionais poderão invocar os recursos educacionais disponíveis no catálogo;

- Conduzir uma avaliação sistemática com especialistas da área de *flipped classroom* com o intuito de validar a abordagem FlipClass-STT, buscando entender se ela cumpre o seu propósito, minimizando o esforço do professor de teste de software que não possui experiência com a implementação do modelo;

- Aprimorar o agente conversacional TOB-STT, visando contemplar aspectos mencionados pelos avaliadores durante o estudo e abordar as outras técnicas e critérios de teste;

- Estabelecer um mecanismo de apoio para apoiar a análise das interações dos alunos com o TOB-STT. Com base nele, o professor poderá identificar, por exemplo, quais tópicos foram responsáveis por gerar a maior quantidade de dúvidas nos alunos.

7.6 Publicações

Durante o período de desenvolvimento desta pesquisa, alguns artigos foram submetidos e publicados em periódicos e congressos. Dentre os estudos estritamente relacionados ao tema:

Periódicos

1. PASCHOAL, L. N.; SOUZA, S. R. S.. **Planejamento e Aplicação de Flipped Classroom para o Ensino de Teste de Software**. *RENOTE. Revista Novas Tecnologias na Educação*, V.16 Nº2, 2018.

Conferências

2. PASCHOAL, L. N.; SILVA, L. R.; SOUZA, S. R. S.. **Abordagem flipped classroom em comparação com o modelo tradicional de ensino: uma investigação empírica no âmbito de Teste de Software**. *In: XXVIII Simpósio Brasileiro de Informática na Educação - SBIE*, Recife, 2017. p. 476-485.

3. PASCHOAL, L. N.; SOUZA, S. R. S.. **A Survey on Software Testing Education in Brazil**. *In: 17th Brazilian Symposium on Software Quality - SBQS*, 2018, Curitiba. p. 334-343.

4. PASCHOAL, L. N.; SOUZA, S. R. S.. **Como Ensinar Teste de Software com Flipped Classroom?**. *In: VIII Workshop on Theses and Dissertations of CBSoft (WTDSoft 2018)*, 2018, São Carlos, SP. p. 46-52.

Outros estudos relacionados à temática da pesquisa também foram publicados no período:

Periódicos

5. PASCHOAL, L. N.; MOZZAQUATRO, P. M. ; FALKEMBACH, G. A. M.. **Concepção, Implementação e Avaliação de um Agente Conversacional com Suporte à Aprendizagem Ubíqua**. *RENOTE. Revista Novas Tecnologias na Educação*, v. 15, p. 01-10, 2017.

Capítulo de livro

6. NUNES, F. B.; HERPICH, F.; PASCHOAL, L. N.. **Virtual Worlds in the Educational Context**. In: Mehdi Khosrow-Pour. (Org.). *Encyclopedia of Information Science and Technology*, Fourth Edition. 4ed.Hershey - Pennsylvania (EUA): IGI Global, 2017, v. 10, p. 7935-7944.

Conferências

7. NUNES, F. B.; HERPICH, F.; PASCHOAL, L. N.; LIMA, J. V.; TAROUCO, L. M. R.. **Mundos Virtuais na prática docente: uma visão sobre os desafios e benefícios**. *In: XXIII Workshop de Informática na Escola*, 2017, Recife, 2017. p. 845-854.

8. PASCHOAL, L. N.; SOUZA, S. R. S.. **Uma Experiência sobre a Aplicação de Aprendizagem Baseada em Projetos com Revisão por Pares no Ensino de Gestão de Sistemas de Informação**. *In: 26º Workshop sobre Educação em Computação*, 2018, Natal - RN. p. 276-285.

9. FIORAVANTI, M. L.; SENA, B.; PASCHOAL, L. N.; SILVA, L. R.; ALLIAN, A. P.; NAKAGAWA, E. Y.; SOUZA, S. R.S.; ISOTANI, S.; BARBOSA, E. F.. **Integrating Project Based Learning and Project Management for Software Engineering Teaching**. *In: 49th ACM Technical Symposium on Computer Science Education*, 2018. p. 806-811.

10. PASCHOAL, L. N.; OLIVEIRA, M. M.; MOZZAQUATRO, P. M.. **A Chatterbot Sensitive to Student's Context to help on Software Engineering Education**. *In: XLIV Conferência Latino-americana de Informática*, 2018, São Paulo, SP, Brasil.

REFERÊNCIAS

ABDELKEFI, M.; KALLEL, I. Conversational agent for mobile-learning: A review and a proposal of a multilanguage text-to-speech agent, "mobispeech". In: **IEEE Tenth International Conference on Research Challenges in Information Science**. [S.l.: s.n.], 2016. p. 1–6. Citado na página 29.

ABDUL-KADER, S. A.; WOODS, J. Survey on chatbot design techniques in speech conversation systems. **International Journal of Advanced Computer Science and Applications**, v. 6, n. 7, p. 72–80, 2015. Citado nas páginas 56, 57 e 58.

ABUSHAWAR, B.; ATWELL, E. Usefulness, localizability, humanness, and language-benefit: additional evaluation criteria for natural language dialogue systems. **International Journal of Speech Technology**, v. 19, n. 2, p. 373–383, 2016. Citado nas páginas 35, 227 e 230.

AGUIAR, E. V. B.; TAROUCO, L. M. R.; REATEGUI, E. Supporting problem-solving in mathematics with a conversational agent capable of representing gifted students' knowledge. In: **47th Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2014. p. 130–137. Citado nas páginas 51, 52 e 211.

ALLISON, M.; JOO, S. F. An adaptive delivery strategy for teaching software testing and maintenance. In: **10th International Conference on Computer Science Education**. [S.l.: s.n.], 2015. p. 237–242. Citado na página 62.

AMIEL, T.; OREY, M.; WEST, R. Recursos educacionais abertos (rea): modelos para localização e adaptação. **ETD - Educação Temática Digital**, v. 12, p. 112–125, nov. 2010. Citado na página 160.

AMRESH, A.; CARBERRY, A. R.; FEMIANI, J. Evaluating the effectiveness of flipped classrooms for teaching cs1. In: **IEEE Frontiers in Education Conference**. [S.l.: s.n.], 2013. p. 733–735. Citado nas páginas 61 e 62.

ANDERSON, L. W.; KRATHWOHL, D. R.; AIRASIAN, P. W. **A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition**. Londres, Reino Unido: Pearson, 2001. Citado na página 152.

ANDRADE, S. A. de. **Execução paralela de programas como suporte ao teste de mutação**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2016. Citado na página 43.

ARIMOTO, M. M.; BARBOSA, E. F. Towards the establishment of an agile method for oers development and delivery. In: **IEEE Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2013. p. 541–547. ISSN 0190-5848. Citado na página 160.

ASTIGARRAGA, T.; DOW, E. M.; LARA, C.; PREWITT, R.; WARD, M. R. The emerging role of software testing in curricula. In: **IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments**. [S.l.: s.n.], 2010. p. 1–26. Citado na página 26.

AşıKSOY, G.; ÖZDAMLı, F. Flipped classroom adapted to the arcs model of motivation and applied to a physics course. **Eurasia Journal of Mathematics, Science and Technology Education**, v. 12, n. 6, p. 1589–1603, 2016. Citado na página 28.

BADA, E. M.; MENEZES, C. S. de. Uma proposta para extração de perguntas e respostas de textos. In: **XVII Congreso Internacional de Informática Educativa**. Santiago, Chile: [s.n.], 2012. Citado na página 29.

BARBOSA, E.; MALDONADO, J. C. Collaborative development of educational modules: A need for lifelong learning. In: MAGOULAS, G. D. (Ed.). **E-Infrastructures and Technologies for Lifelong Learning: Next Generation Environments**. Hershey, EUA: IGI Global, 2011. cap. 7, p. 175–211. Citado na página 45.

BARBOSA, E. F. **Uma Contribuição ao Processo de Desenvolvimento e Modelagem de Módulos Educacionais**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2004. Citado na página 156.

BARBOSA, E. F.; CHAIM, M. L.; DELAMARO, M. E.; JINO, M.; MALDONADO, J. C. Teste estrutural. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). **Introdução ao Teste de Software**. Rio de Janeiro, Brasil: Elsevier, 2016. cap. 4, p. 59–92. Citado nas páginas 42 e 155.

BARBOSA, E. F.; NAKAGAWA, E. Y.; MALDONADO, J. C. Towards the establishment of an ontology of software testing. In: **18th International Conference on Software Engineering & Knowledge Engineering**. San Francisco, USA: [s.n.], 2006. Citado na página 143.

BARBOSA, E. F.; SOUZA, S. R. S.; DOMINGUES, A. L. S.; CHAN, A.; HöHN, E. N.; MALDONADO, J. C. Uma experiência no ensino de inspeção e teste de software. In: **VI Simpósio Brasileiro de Qualidade de Software**. Recife, Pernambuco, Brasil: [s.n.], 2007. p. 309–324. Citado na página 26.

BARBOSA, J. R.; VALLE, P. H.; MALDONADO, J.; DELAMARO, M.; VINCENZI, A. M. R. An experimental evaluation of peer testing in the context of the teaching of software testing. In: **19th International Symposium on Computers in Education**. Lisbon, Portugal: [s.n.], 2017. p. 01–06. Citado na página 46.

BARROS, F.; TEDESCO, P. Agentes inteligentes conversacionais: Conceitos básicos e desenvolvimento. In: **35ª Jornada de Atualização em Informática - XXXVI Congresso da Sociedade Brasileira de Computação**. Porto Alegre, Rio Grande do Sul, Brasil: [s.n.], 2016. Citado nas páginas 57 e 212.

BASILI, V. R. **Software Modeling and Measurement: The Goal/Question/Metric Paradigm**. College Park, MD, USA, 1992. Disponível em: <https://drum.lib.umd.edu/bitstream/handle/1903/7538/?sequence=1>. Acesso em: 19 fev. 2018. Citado na página 143.

BASILI, V. R.; ROMBACH, H. D. The tame project: towards improvement-oriented software environments. **IEEE Transactions on Software Engineering**, v. 14, n. 6, p. 758–773, 1988. Citado na página 93.

BENITTI, F. B. V. Avaliando objetos de aprendizagem para o ensino de teste de software. In: **XX Conferência Internacional sobre Informática na Educação**. Santiago, Chile: [s.n.], 2015. p. 584–589. Citado na página 161.

BENITTI, F. B. V.; ALBANO, E. L. Teste de software: o que e como é ensinado? In: **XX Workshop sobre Educação em Computação**. Paraná, Brasil: [s.n.], 2012. p. 01–10. Citado na página 147.

BERGMANN, J.; SAMS, A. **Flip your classroom: Reach every student in every class every day**. Washington, D.C., EUA: International Society for Technology in Education, 2012. Citado nas páginas 28, 30, 33, 34, 47, 48, 63, 89, 118, 121, 126, 139, 141, 151, 156, 167 e 178.

BERTOLINO, A.; MARCHETTI, E. Software testing. In: BOURQUE, P.; FAIRLEY, R. E. (Ed.). **SWEBOK: Guide to the Software Engineering Body of Knowledge**. Washington, EUA: IEEE Computer Society, 2014. p. 82–103. Citado nas páginas 143 e 212.

BICKMORE, T. W.; PFEIFER, L. M.; BYRON, D.; FORSYTHE, S.; HENAULT, L. E.; JACK, B. W.; SILLIMAN, R.; PAASCHE-ORLOW, M. K. Usability of conversational agents by patients with inadequate health literacy: Evidence from two clinical trials. **Journal of Health Communication**, v. 15, n. 2, p. 197–210, 2010. Citado na página 52.

BIGHAM, J. P.; ALLER, M. B.; BRUDVIK, J. T.; LEUNG, J. O.; YAZZOLINO, L. A.; LADNER, R. E. Inspiring blind high school students to pursue computer science with instant messaging chatbots. In: **39th ACM Technical Symposium on Computer Science Education**. [S.l.: s.n.], 2008. p. 449–453. Citado na página 52.

BISHOP, J.; VERLEGER, M. A. The flipped classroom: A survey of the research. In: **120th ASEE Annual Conference & Exposition**. Atlanta, Georgia: [s.n.], 2013. p. 1–18. Citado na página 33.

BOS, A. S. **A.G.I.M.C: Agente Inteligente Conversacional Como Guia Em Um Museu Virtual 3D da Computação**. Dissertação (Mestrado) — Centro de Tecnologia - Universidade Federal de Santa Maria, Santa Maria, 2014. Citado na página 53.

BRADEšKO, L.; MLADENIć, D. A survey of chatbot systems through a loebner prize competition. In: **8th Language Technologies Conference**. Ljubljana, Slovenia: [s.n.], 2012. p. 01–04. Citado nas páginas 56 e 58.

BRADY, F. **Cambridge University Study States Software Bugs Cost Economy $312 Billion Per Year**. 2013. Disponível em: <http://markets.financialcontent.com/stocks/news/read/23147130/Cambridge_University_Study_States_Software_Bugs_Cost_Economy_\protect\TU\textdollar312_Billion_Per_Year>. Acesso em: 29/11/2017. Citado na página 25.

BUSSAB, W. O.; MORETTIN, P. A. **Estatística Básica**. São Paulo: Saraiva, 2013. Citado na página 144.

BUTCHER, N. **A basic guide to open educational resources (OER)**. [S.l.]: Commonwealth of Learning (COL);, 2015. Citado na página 160.

CARVALHO, R. J. O.; RAMOS, M. A. S. Flipped classroom - centrar a aprendizagem no aluno recorrendo a ferramentas cognitivas. In: **IX International Conference on ICT in Education**. Porto, Portugal: [s.n.], 2015. Citado na página 27.

CHAIM, M. L. **Poke-Tool - uma ferramenta para suporte ao teste estrutural de programas baseados em análise de fluxo de dados**. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1991. Citado na página 156.

CHAN, F. T.; TSE, T. H.; TANG, W. H.; CHEN, T. Y. Software testing education and training in hong kong. In: **5th International Conference on Quality Software**. Melbourne, Victoria, Australia: [s.n.], 2005. Citado na página 26.

CLARK, N. Peer testing in software engineering projects. In: **6th Australasian Conference on Computing Education**. Darlinghurst, Australia: [s.n.], 2004. p. 41–48. Citado na página 46.

CLARKE, P. J.; DAVIS, D.; KING, T. M.; PAVA, J.; JONES, E. L. Integrating testing into software engineering courses supported by a collaborative learning environment. **ACM Transactions on Computing Education**, ACM, New York, NY, USA, v. 14, n. 3, p. 18:1–18:33, 2014. Citado na página 26.

CLARKE, P. J.; PAVA, J.; WU, Y.; KING, T. M. Collaborative web-based learning of testing tools in se courses. In: **42nd ACM Technical Symposium on Computer Science Education**. Dallas, TX, USA: [s.n.], 2011. Citado na página 45.

CLEGG, B. S.; ROJAS, J. M.; FRASER, G. Teaching software testing concepts using a mutation testing game. In: **39th International Conference on Software Engineering**. Buenos Aires, Argentina: [s.n.], 2017. p. 33–36. Citado na página 25.

COHEN, J. A power primer. **Psychological bulletin**, American Psychological Association, v. 112, n. 1, p. 155, 1992. Citado na página 134.

COOK, D. A.; BECKMAN, T. J. Current concepts in validity and reliability for psychometric instruments: Theory and application. **The American Journal of Medicine**, v. 119, n. 2, p. 166.e7 – 166.e16, 2006. Citado na página 146.

COOK, T. D.; CAMPBELL, D. T. **Quasi-Experimentation: Design & Analysis Issues for Field Settings**. Boston, Massachusetts, United States: Houghton Mifflin, 1979. Citado nas páginas 89 e 136.

CRAIG, S. D.; SCHROEDER, N. L. Reconsidering the voice effect when learning from a virtual human. **Computer & Education**, v. 114, n. 1, p. 193–205, 2017. Citado na página 52.

CRESPO, A. N. **Modelos de Confiabilidade de Software Baseados em Cobertura de Critérios Estruturais de Teste**. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1997. Citado nas páginas 41, 42 e 43.

CRONBACH, L. J. Coefficient alpha and the internal structure of tests. **Psychometrika**, v. 16, n. 3, p. 297–334, Sep 1951. Citado nas páginas 233 e 238.

CURY, A. **Organização e métodos: uma visão holística**. [S.l.]: Atlas, 2006. Citado na página 123.

DAVIES, R. S.; DEAN, D. L.; BALL, N. Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course. **Educational Technology Research and Development**, v. 61, n. 4, p. 563–580, 2013. Citado na página 27.

DELAMARO, M.; JINO, M.; MALDONADO, J. **Introdução ao Teste de Software**. Rio de Janeiro, Brasil: Elsevier, 2016. Citado na página 151.

DELAMARO, M. E. **Proteum: Um Ambiente de Teste Baseado na Análise de Mutantes**. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, 1993. Citado na página 44.

DELAMARO, M. E. **Mutação de Interface: Um Critério de Adequação Interprocedimental para o Teste de Integração**. Tese (Doutorado) — Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 1997. Citado nas páginas 41 e 44.

DELAMARO, M. E.; MALDONADO, J. C. Proteum/im 2.0: An integrated mutation testing environment. In: WONG, W. E. (Ed.). **Mutation Testing for the New Century**. Norwell, MA, USA: Kluwer Academic Publishers, 2001. cap. 14, p. 91–101. Citado na página 156.

DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. Conceitos básicos. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). **Introdução ao Teste de Software**. Rio de Janeiro, Brasil: Elsevier, 2016. cap. 1, p. 1–8. Citado nas páginas 26, 38, 39 e 162.

DELAMARO, M. E.; OLIVEIRA, R. A. O.; BARBOSA, E. F.; MALDONADO, J. C. Teste de mutação. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). **Introdução ao Teste de Software**. Rio de Janeiro, Brasil: Elsevier, 2016. cap. 5, p. 93–142. Citado na página 44.

DEMILLO, R. A.; LIPTON, R. J.; SAYWARD, F. G. Hint on test data selection: Help for the practicing programmer. **Computer**, v. 11, n. 4, p. 34–41, 1980. Citado nas páginas 43 e 44.

DERYUGINA, O. V. Chatterbots. **Scientific and Technical Information Processing**, v. 37, n. 2, p. 143–147, Apr 2010. Citado nas páginas 29 e 51.

DINIZ, L. L.; DAZZI, R. L. S. Jogo para o apoio ao ensino de teste de caixa-preta. In: **XXII Simpósio Brasileiro de Informática na Educação**. Aracaju, Brasil: [s.n.], 2011. p. 426–435. Citado na página 155.

DUTRA, R.; TAROUCO, L.; PASSERINO, L. Utilização de objetos de aprendizagem abertos scorm para dar suporte à avaliação formativa. **Revista Brasileira de Informática na Educação**, v. 18, n. 03, 2011. Citado na página 174.

FABBRI, S. C. P. F.; VINCENZI, A. M. R.; FREITAS, E. N. A.; MALDONADO, J. C. Teste estrutural. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). **Introdução ao Teste de Software**. Rio de Janeiro, Brasil: Elsevier, 2016. cap. 2, p. 9–38. Citado nas páginas 41 e 137.

FARIAS, V.; MOREIRA, C.; COUTINHO, E.; SANTOS, I. S. itest learning: um jogo para o ensino do planejamento de testes de software. In: **V Fórum de Educação em Engenharia de Software**. Natal, Brasil: [s.n.], 2012. Citado na página 155.

FASSBINDER, A. G. de O.; BOTELHO, T. G.; MARTINS, R. J.; BARBOSA, E. F. Applying flipped classroom and problem-based learning in a cs1 course. In: **45th Annual Frontiers in Education Conference**. El Paso, TX, USA: [s.n.], 2015. Citado nas páginas 27, 48 e 180.

FASSBINDER, A. G. de O.; MOREIRA, D.; CRUZ, G.; BARBOSA, E. F. Tools for the flipped classroom model: An experiment in teacher education. In: **44th Annual Frontiers in Education Conference**. Madrid, Spain: [s.n.], 2014. p. 1–8. Citado nas páginas 28, 47, 152, 154, 155 e 156.

FELDER, R. M.; SILVERMAN, L. K. Learning and teaching styles in engineering education. **International Journal of Engineering Education**, v. 78, n. 7, p. 674–681, 1998. Citado na página 155.

FERLIN, J.; KEMCZINSKI, A.; MURAKAMI, E.; HOUNSELL, M. da S. Metadados essenciais: Uma metodologia para catalogação de objetos de aprendizagem no repositório digital roai. In: **I Workshop de Informática na Escola**. [S.l.: s.n.], 2010. p. 1147–1156. Citado na página 174.

FERRAZ, A. P. do C. M.; BELHOT, R. V. Taxonomia de bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. **Gestão & Produção**, v. 17, n. 2, p. 421 – 431, 2010. Citado na página 48.

FIGUEREDO, C. O.; SANTOS, S. C.; BORBA, P. H.; ALEXANDRE, G. H. Using pbl to develop software test engineers. In: **14th International Conference on Computers and Advanced Technology in Education**. Cambridge, United Kingdow: [s.n.], 2011. p. 1–7. Citado na página 45.

FRANKL, P. G.; WEYUKER, E. J. Testing software to detect and reduce risk. **Journal of Systems and Software**, v. 53, n. 3, p. 275–286, 2000. Citado na página 40.

FULTON, K. Upside down and inside out: Flip your classroom to improve student learning. **Learning & Leading with Technology**, v. 39, n. 8, p. 12–17, 2012. Citado na página 50.

GANNOD, G.; BURGE, J.; HELMICK, M. Using the inverted classroom to teach software engineering. In: **30th International Conference on Software Engineering**. Leipzig, Germany: [s.n.], 2008. p. 777–786. Citado nas páginas 28, 93 e 163.

GEHRINGER, E. F.; PEDDYCORD III, B. W. The inverted-lecture model: A case study in computer architecture. In: **44th ACM Technical Symposium on Computer Science Education**. New York, NY, USA: ACM, 2013. p. 489–494. Citado na página 27.

GHOSE, S.; BARUA, J. J. Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor. In: **2nd International Conference on Informatics, Electronics & Vision**. Dhaka, Bangladesh: [s.n.], 2013. p. 1–5. Citado na página 29.

GIANNAKOS, M. N.; KROGSTIE, J.; CHRISOCHOIDES, N. Reviewing the flipped classroom research: Reflections for computer science education. In: **Computer Science Education Research Conference**. [S.l.: s.n.], 2014. p. 23–29. Citado nas páginas 27 e 61.

GILBOY, M. B.; HEINERICHS, S.; PAZZAGLIA, G. Enhancing student engagement using the flipped classroom. **Journal of Nutrition Education and Behavior**, v. 47, n. 1, p. 109 – 114, 2015. Citado nas páginas 47, 48, 49 e 163.

GORDILLO, A.; BARRA, E.; QUEMADA, J. An easy to use open source authoring tool to create effective and reusable learning objects. **Computer Applications in Engineering Education**, v. 25, n. 2, p. 188–199, 2017. Citado na página 161.

GRAESSER, A. C.; CAI, Z.; MORGAN, B.; WANG, L. Assessment with computer agents that engage in conversational dialogues and trialogues with learners. **Computers in Human Behavior**, v. 76, p. 607–616, 2017. Citado na página 52.

HAUCK, J. C. R.; WANGENHEIM, C. G. von; WANGENHEIM, A. von. **Método de Aquisição de Conhecimento para Customização de Modelos de Capacidade/-Maturidade de Processos de Software**. Florianópolis, Santa Catarina, Brasil, 2011. Disponível em: <https://goo.gl/4mEiTg>. Acesso em: 19 fev. 2018. Citado na página 146.

HE, W.; HOLTON, A.; FARKAS, G.; WARSCHAUER, M. The effects of flipped instruction on out-of-class study time, exam performance, and student perceptions. **Learning and Instruction**, v. 45, p. 61 – 71, 2016. Citado nas páginas 92, 93, 98, 100, 110, 112, 115, 117, 119, 183 e 185.

HERPICH, F.; NUNES, F. B.; VOSS, G. B.; MEDINA, R. D. Three-dimensional virtual environment and npc: A perspective about intelligent agents ubiquitous. In: NETO, F. M. M.; SOUZA, R. de; GOMES, A. S. (Ed.). **Handbook of Research on 3-D Virtual Environments and Hypermedia for Ubiquitous Learning**. Hershey, EUA: IGI Global, 2016. cap. 21, p. 510–536. Citado nas páginas 29, 35, 52, 54 e 232.

HERREID, C. F.; SCHILLER, N. A. Case studies and the flipped classroom. **Journal of College Science Teaching**, v. 42, n. 5, p. 62–66, 2013. Citado na página 50.

HETZEL, W. O que é teste de software: Introdução. In: HETZEL, W. (Ed.). **Guia Completo ao Teste de Software**. Rio de Janeiro, Brasil: Campus LTDA, 1987. cap. 1, p. 3–17. Citado na página 38.

HOWDEN, W. E. **Functional Program Testing and Analysis**. New York, NY, USA: McGraw-Hill, Inc., 1986. Citado na página 41.

ISOTANI, S.; BITTENCOURT, I. I. **Dados Abertos Conectados: Em busca da Web do Conheciment**. São Paulo: Novatec Editora, 2015. Citado nas páginas 29 e 58.

JAYAL, A. **Framework to manage labels for e-assessment of diagrams**. Tese (Doutorado) — Brunel University, School of Information Systems, Computing and Mathematics, 2010. Citado na página 174.

JORGENSEN, P. C. **Software testing: a craftsman's approach**. [S.l.]: Auerbach Publications, 2013. Citado na página 137.

KASUNIC, M. **Designing an effective survey**. Pittsburgh, Pensilvânia, EUA, 2005. Citado nas páginas 141, 142, 143 e 146.

KELLER, J. M. **Motivational design for learning and performance: The ARCS model approach**. [S.l.]: Springer Science & Business Media, 2009. Citado nas páginas 68, 69 e 73.

KERR, B. The flipped classroom in engineering education: A survey of the research. In: 18TH INTERNATIONAL CONFERENCE ON INTERACTIVE COLLABORATIVE LEARNING. Florence, Italy, 2015. p. 815–818. Citado na página 27.

KHALSA, S. K.; LABICHE, Y.; NICOLETTA, J. The power of single and error annotations in category partition testing: An experimental evaluation. In: **20th International Conference on Evaluation and Assessment in Software Engineering**. New York, NY, USA: ACM, 2016. p. 28:1–28:10. Citado na página 137.

KIAT, P. N.; KWONG, Y. T. The flipped classroom experience. In: **27th Conference on Software Engineering Education and Training**. [S.l.: s.n.], 2014. p. 39–43. Citado na página 27.

KIRSCHNER, P. A. Stop propagating the learning styles myth. **Computers & Education**, v. 106, p. 166 – 171, 2017. Citado na página 155.

KITCHENHAM, B. A.; PFLEEGER, S. L. Personal opinion surveys. In: SHULL, F.; SINGER, J.; SJØBERG, D. I. K. (Ed.). **Guide to Advanced Empirical Software Engineering**. London: Springer London, 2008. p. 63–92. Citado na página 141.

KITCHENHAM, B. A.; PFLEEGER, S. L.; PICKARD, L. M.; JONES, P. W.; HOAGLIN, D. C.; EMAM, K. E.; ROSENBERG, J. Preliminary guidelines for empirical research in software engineering. **IEEE Transactions on Software Engineering**, v. 28, n. 8, p. 721–734, 2002. Citado na página 28.

KIWELEKAR, A. W.; WANKHEDE, H. S. Learning objectives for a course on software architecture. In: WEYNS, D.; MIRANDOLA, R.; CRNKOVIC, I. (Ed.). **Software Architecture**. Cham: Springer International Publishing, 2015. p. 169–180. Citado nas páginas 151 e 153.

KONG, S. C. Developing information literacy and critical thinking skills through domain knowledge learning in digital classrooms: An experience of practicing flipped classroom strategy. **Computers & Education**, v. 78, p. 160 – 173, 2014. Citado na página 92.

KOSTARIS, C.; SERGIS, S.; SAMPSON, D. G.; GIANNAKOS, M. Í.; PELLICCIONE, L. Investigating the potential of the flipped classroom model in k-12 ict teaching and learning: An action research study. **Journal of Educational Technology & Society**, JSTOR, v. 20, n. 1, p. 261–273, 2017. Citado nas páginas 28 e 68.

LAND, L. P. W.; HIGGS, J. An empirical study of software quality improvement practices from multiple perspectives – an australian case study. In: **11th Pacific-Asia Conference on Information Systems**. Auckland, New Zealand: [s.n.], 2007. p. 547–560. Citado na página 25.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **Biometrics**, [Wiley, International Biometric Society], v. 33, n. 1, p. 159–174, 1977. Citado nas páginas 234 e 239.

LAUVåS, P.; ARCURI, A. Recent trends in software testing education: A systematic literature review. In: **The Norwegian Conference on Didactics in IT education**. [S.l.: s.n.], 2018. p. 1–11. Citado na página 32.

LEMOS, E. das C. **Desenvolvimento de Chatterbots Educacionais: um estudo de caso voltado ao ensino de Algoritmos**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Bambuí, 2011. Citado na página 53.

LEMOS, O. A. L.; FERRARI, F. C.; SILVEIRA, F. F.; GARCIA, A. Experience report: Can software testing education lead to more reliable code? In: **26th International Symposium on Software Reliability Engineering**. Gaithersbury, MD, USA: [s.n.], 2015. p. 359–369. Citado nas páginas 26 e 62.

LEMOS, O. A. L.; SILVEIRA, F. F.; FERRARI, F. C.; GARCIA, A. The impact of software testing education on code reliability: An empirical assessment. **Journal of Systems and Software**, v. 137, n. 1, p. 497–511, 2018. Citado na página 62.

LEONHARDT, M. D.; CASTRO, D. D. de; DUTRA, R. L. de S.; TAROUCO, L. M. R. Elektra: Um chatterbot para uso em ambiente educacional. **RENOTE - Revista Novas Tecnologias na Educação**, v. 1, n. 2, p. 01–10, 2003. Citado nas páginas 29 e 211.

LEONHARDT, M. D.; NEISSE, R.; TAROUCO, L. M. R. Meara: Um chatterbot temático para uso em ambiente educacional. In: **XIV Simpósio Brasileiro de Informática na Educação**. Rio de Janeiro, Brasil: [s.n.], 2003. p. 81–88. Citado na página 29.

LEONHARDT, M. D.; TAROUCO, L.; VICARI, R. M.; SANTOS, E. R.; SILVA, M. dos Santos da. Using chatbots for network management training through problem-based oriented education. In: **7th International Conference on Advanced Learning Technologies**. Niigata, Japan: [s.n.], 2007. p. 845–847. Citado na página 52.

LIN, L.; ATKINSON, R. K.; CHRISTOPHERSON, R. M.; JOSEPH, S. S.; HARRISON, C. J. Animated agents and learning: Does the type of verbal feedback they provide matter? **Computer & Education**, v. 67, n. 1, p. 239–249, 2013. Citado na página 52.

LITWIN, M. S. **How to Measure Survey Reliability and Validity**. Thousand Oaks, California: SAGE Publication, 1995. Citado na página 146.

LOVE, B.; HODGE, A.; GRANDGENETT, N.; SWIFT, A. W. Student learning and perceptions in a flipped linear algebra course. **International Journal of Mathematical Education in Science and Technology**, Taylor & Francis, v. 45, n. 3, p. 317–324, 2014. Citado na página 61.

LUCCHESI, I. L.; SILVA, A. R. da; ABREU, C.; TAROUCO, L. M. R. Avaliação de um chatbot no contexto educacional: Um relato de experiência com metis. **RENOTE - Revista Novas Tecnologias na Educação**, v. 16, n. 1, p. 1–10, 2018. Citado na página 211.

LYRA, K. T.; ISOTANI, S.; REIS, R. C. D.; MARQUES, L. B.; PEDRO, L. Z.; JAQUES, P. A.; BITENCOURT, I. I. Infographics or graphics+text: Which material is best for robust learning? In: **16th International Conference on Advanced Learning Technologies**. Austin, TX, USA: [s.n.], 2016. p. 366–370. Citado nas páginas 96, 105 e 155.

MAHER, M. L.; LATULIPE, C.; LIPFORD, H.; RORRER, A. Flipped classroom strategies for cs education. In: **46th ACM Technical Symposium on Computer Science Education**. New York, NY, USA: ACM, 2015. p. 218–223. Citado na página 61.

MALDONADO, J.; DELAMARO, M.; VINCENZI, A. **Automatização de teste de software com ferramentas de software livre**. [S.l.]: Elsevier Editora Ltda., 2018. Citado na página 162.

MALDONADO, J. C. **Critérios Potenciais Usos: Uma Contribuição ao Teste Estrutural de Software**. Tese (Doutorado) — Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 1991. Citado na página 38.

MALDONADO, J. C.; VINCENZI, A. M. R.; BARBOSA, E. F.; SOUZA, S. R. S.; DELAMARO, M. E. Aspectos teóricos e empíricos de teste de cobertura de software. In: **VI Escola de Informática da SBC - Regional Sul**. Blumenau, Santa Catarina, Brasil: [s.n.], 1998. p. 53–86. Citado na página 40.

MANFIO, E. R.; MORENO, F. A evolução dos chatterbots: Pln, ia e difusão cultural. **Revista e-f@tec**, v. 5, n. 1, p. 01–08, 2015. Citado na página 51.

MASSARO, D. W.; OUNI, S.; COHEN, M.; CLARK, R. A multilingual embodied conversational agent. In: **38th Annual Hawaii International Conference on System Science**. Big Island, HI, USA: [s.n.], 2005. p. 1–8. Citado na página 29.

MAYER, R. E. Rote versus meaningful learning. **Theory Into Practice**, Routledge, v. 41, n. 4, p. 226–232, 2002. Citado na página 117.

MCNEAL, M. L.; NEWYEAR, D. Chatbot creation options. **Library Technology Reports**, v. 49, n. 8, p. 11–17, 2013. Citado na página 57.

MEDINA, R. D. **ASTERIX - Aprendizagem Significativa e Tecnologias aplicadas no Ensino de Redes de computadores- Integrando e eXplorando possibilidades**. Tese (Doutorado) — Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, Brasil, 2004. Citado na página 182.

MELO, S. M. **Avaliação do custo e efetividade dos critérios de teste estruturais no contexto de programas concorrentes com memória compartilhada**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012. Citado na página 40.

MELO, S. M. **Um framework para avaliação sistemática de técnicas de teste no contexto de programação concorrente**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2018. Citado na página 182.

MIZOGUCHI, R. Tutorial on ontological engineering: part 3: Advanced course of ontological engineering. **New Generation Computing**, v. 22, n. 2, p. 198–220, 2004. Citado nas páginas 58 e 59.

MORAES, S.; SANTOS, A.; REDECKER, M. Prototipação de chatterbots como método de aprendizagem em cursos de computação: uma experiência em sala de aula. In: **XXII Workshop de Informática na Escola**. Uberlândia, Minas Gerais, Brasil: [s.n.], 2016. p. 91–100. Citado na página 29.

MYERS, G. J.; SANDLER, C.; BADGETT, T. **The Art of Software Testing**. Hoboken, Nova Jersey, EUA: Wiley Publishing, 2011. Citado nas páginas 25 e 162.

NAKAGAWA, E. Y. **Uma Contribuição ao Projeto Arquitetural de Ambientes de Engenharia de Software**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2006. Citado nas páginas 40 e 41.

NAPS, T.; COOPER, S.; KOLDEHOFE, B.; LESKA, C.; RößLING, G.; DANN, W.; KORHONEN, A.; MALMI, L.; RANTAKOKKO, J.; ROSS, R. J.; ANDERSON, J.; FLEISCHER, R.; KUITTINEN, M.; MCNALLY, M. Evaluating the educational impact of visualization. In: **Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education**. New York, NY, USA: ACM, 2003. p. 124–136. Citado na página 155.

NEVES, A. M. M. **iAIML: um mecanismo para o tratamento de intenção em Chatterbots**. Tese (Doutorado) — Universidade Federal de Pernambuco, Recife, Pernambuco, Brasil, 2005. Citado na página 56.

OLIVEIRA, B. C. **TestEG - um software educacional para o ensino de teste de software**. Monografia (Graduação) — Universidade Federal de Lavras, Lavras, 2013. Citado na página 155.

OLIVEIRA, H. T. A. de; GADELHA, R. N. S.; AZEVEDO, R. R. de; JúNIOR, J. B. D.; DIAS, G. A.; FREITAS, F. Dr. pierre: Um chatterbot com intenção e personalidade baseado em ontologias para apoiar o ensino de psiquiatria. In: **XXII Simpósio Brasileiro de Informática na Educação**. Aracajú - Sergipe: [s.n.], 2011. p. 01–10. Citado na página 58.

PASCHOAL, L. N.; CHICON, P. M. M.; FALKEMBACH, G. A. M. Concepção, implementação e avaliação de um agente conversacional com suporte à aprendizagem ubíqua. **RENOTE - Revista Novas Tecnologias na Educação**, v. 15, n. 1, p. 01–10, 2017. Citado nas páginas 29 e 52.

PASCHOAL, L. N.; MOZZAQUATRO, P. M.; KRASSMANN, A. L.; BINELO, M. O. Ubibot: Agente inteligente consciente do contexto de aprendizagem do usuário integrado ao ambiente moodle. In: **XXI Congreso Internacional de Informática Educativa**. Santiago, Chile: [s.n.], 2016. p. 95 – 104. Citado na página 29.

PASCHOAL, L. N.; OLIVEIRA, M. M. de; CHICON, P. M. M. A chatterbot sensitive to student's context to help on software engineering education. In: **XLIV Conferencia Latinoamericana de Informática**. [S.l.: s.n.], 2018. p. 1–10. Citado nas páginas 35, 51, 52, 55, 211 e 232.

PASCHOAL, L. N.; SILVA, L. R.; SOUZA, S. R. S. Abordagem flipped classroom em comparação com o modelo tradicional de ensino: uma investigação empírica no Âmbito de teste de software. In: **XXVIII Simpósio Brasileiro de Informática na Educação**. Recife, Pernambuco, Brasil: [s.n.], 2017. p. 476–485. Citado nas páginas 26, 27, 28, 46 e 92.

PASCHOAL, L. N.; SOUZA, S. R. Planejamento e aplicação de flipped classroom para o ensino de teste de software. **RENOTE - Revista Novas Tecnologias na Educação**, v. 16, n. 2, p. 1–10, 2018. Citado nas páginas 27, 29, 106, 119, 137 e 169.

PASCHOAL, L. N.; SOUZA, S. R. S. A survey on software testing education in brazil. In: **17th Brazilian Symposium on Software Quality**. New York, NY, USA: ACM, 2018. p. 334–343. Citado nas páginas 62, 64, 91, 119, 137 e 169.

PAZ, F. J.; SILVEIRA, C.; KRASSMANN, A.; TAROUCO, L. M. R. Perspectivas tecnológicas para o aprimoramento de chatbots educacionais em aiml. **Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación**, v. 20, n. 1, p. 7–15, 2017. Citado na página 57.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **12th International Conference on Evaluation and Assessment in Software Engineering**. Italy: [s.n.], 2008. p. 68–77. Citado na página 35.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, n. 1, p. 1 – 18, 2015. Citado na página 35.

POLSANI, P. Use and abuse of reusable learning objects. **Journal of Digital Information**, v. 3, n. 4, p. 01–05, 2003. Citado na página 161.

PRADO, M. P. **Um estudo de caracterização e avaliação de critérios de teste estruturais entre os paradigmas procedimental e OO**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2009. Citado na página 42.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. São Paulo, Brasil: McGraw Hill Brasil, 2016. Citado nas páginas 37 e 55.

PRIMO, A. F. T.; COELHO, L. R.; PAIM, M. F. R.; REICHEL, D. O uso de chatterbots na educação à distânci. In: **IV Workshop de Informática na Educação.** Porto Alegre, Rio Grande do Sul, Brasil: [s.n.], 1998. p. 1–9. Citado na página 29.

RADZIWILL, N. M.; BENTON, M. C. Evaluating quality of chatbots and intelligent conversational agents. In: **arXiv:1704.04579 [cs.CY]**. Ithaca, New York, USA: [s.n.], 2017. Citado na página 52.

RAMíREZ, D.; HINOJOSA, C.; RODRÍGUEZ, F. Advantages and disadvantages of flipped classroom: Stem students' perception. In: **7th International Conference of Education, Research and Innovation**. Seville, Spain: [s.n.], 2014. p. 121–127. Citado nas páginas 28 e 50.

RAPPS, S.; WEYUKER, E. Selecting software test data using data flow information. **IEEE Transactions on Software Engineering**, v. 11, n. 4, p. 367–375, 1985. Citado na página 42.

RIBEIRO, P. da S.; FRANCISCATO, F. T.; MOZZAQUATRO, P. M.; MEDINA, R. D. Repositório de vídeos educacionais adaptado as necessidades tecnológicas de usuários de dispositivos móveis. In: **XVI Workshop Sobre Informática na Escola**. Belo Horizonte, Minas Gerais, Brasil: [s.n.], 2010. p. 1265–1274. Citado na página 29.

RIOS, E.; MOREIRA, T. **Teste de Software**. Rio de Janeiro: Alta Books, 2013. Citado nas páginas 54 e 55.

ROJAS, J. M.; WHITE, T. D.; CLEGG, B. S.; FRASER, G. Code defenders: Crowdsourcing effective tests and subtle mutants with a mutation testing game. In: **39th International Conference on Software Engineering**. Buenos Aires, Argentina: [s.n.], 2017. p. 677–688. Citado nas páginas 45 e 155.

RONDON-MELO, S.; ANDRADE, C. R. F. d. Educação mediada por tecnologia em fonoaudiologia: impacto na motivação para aprendizagem sobre o sistema miofuncional orofacial. **CoDAS**, v. 28, n. 3, p. 269 – 277, 2016. Citado na página 79.

ROSA, L. H. C.; LEMOS, E.; BERNARDI, G.; MEDINA, R. Ensino de testes de software por meio de digital storytelling e chatterbots. In: **XXVIII Simpósio Brasileiro de Informática na Educação**. Recife, Pernambuco, Brasil: [s.n.], 2017. p. 797–806. Citado nas páginas 29 e 54.

SANTANA, B.; ARAÚJO, L. G.; BITTENCOURT, R. Considerando a motivação dos estudantes em experiências de ensino-aprendizagem de computação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2018. p. 500. Citado nas páginas 75 e 91.

SANTOS, L. M. A. **A inserção de um agente conversacional animado em um ambiente virtual de aprendizagem a partir da teoria da carga cognitiva**. Tese (Doutorado) — Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, Brasil, 2009. Citado nas páginas 51, 52 e 211.

SCHMITZ, E. X. S. **Sala de aula invertida: uma abordagem para combinar metodologias ativas e engajar alunos no processo de ensino-aprendizagem**. 01 - 187 p. Dissertação (Mestrado) — Universidade Federal de Santa Maria, Centro de Educação, 2016. Citado na página 47.

SCHNEIDER, C. K.; CAETANO, L.; RIBEIRO, L. O. M. Análise de vídeos educacionais no youtube: caracteres e legibilidade. **Renote**, v. 10, n. 1, 2012. Citado na página 157.

SHAIKH, A.; PHALKE, G.; PATIL, P.; BHOSALE, S.; RAGHATWAN, J. Survey on chatbot design techniques in speech conversation systems. **Journal of Advanced Computer Science and Applications**, v. 6, n. 7, 2015. Citado na página 29.

SHIPPEY, T. J. **Exploiting Abstract Syntax Trees to Locate Software Defects**. Tese (Doutorado) — University of Hertfordshire, School of Computer Sciences, United Kingdom, 2015. Citado na página 25.

SILVA, J. H.; COUTO, L. S.; ODAKURA, V. Aparência de um agente pedagógico animado para um ambiente virtual de aprendizagem. In: **XXIII Simpósio Brasileiro de Informática na Educação**. Rio de Janeiro, Brasil: [s.n.], 2012. p. 1–5. Citado na página 29.

SILVA, J. M. C. **Análise Técnica e Pedagógica de Metadados para Objetos de Aprendizagem**. 01 - 187 p. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Centro Interdisciplinar de Novas Tecnologias na Educação, 2011. Citado na página 161.

SILVA, T. G. **Jogo Sério em Mundos Virtuais: uma abordagem para ensino-aprendizagem de Teste de Software**. Dissertação (Mestrado) — Centro de Tecnologia - Universidade Federal de Santa Maria, Santa Maria, 2012. Citado na página 155.

SILVA, T. G. da; MüLLER, F. M.; BERNARDI, G. Panorama do ensino de engenharia de software em cursos de graduação focado em teste de software: Uma proposta de aprendizagem baseada em jogos. **RENOTE - Revista Novas Tecnologias na Educação**, v. 9, n. 2, p. 01–10, 2011. Citado na página 147.

SILVA, V. M.; BARBOSA, E. F.; FASSBINDER, A. G. O. Ferramenta web de apoio à validação de mapas de aprendizagem para moocs. In: **Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2017. p. 284–291. Citado na página 156.

SILVEIRA, J. L.; THIRY, M. Desenvolvimento de um chatterbot para Área de medição de software. In: **I Computer on the Beach**. Florianópolis, Santa Catarina, Brasil: [s.n.], 2010. Citado na página 52.

SINGH, B.; GAUTAM, S. The impact of software development process on software quality: A review. In: **8th International Conference on Computational Intelligence and Communication Networks (CICN)**. Tehri, India: [s.n.], 2016. p. 666–672. Citado na página 25.

SMITH, J.; TESSLER, J.; KRAMER, E.; LIN, C. Using peer review to teach software testing. In: **9th Annual International Conference on International Computing Education Research**. Auckland, New Zealand: [s.n.], 2012. p. 93–98. Citado nas páginas 26, 45 e 46.

SOMMERVILLE, I. Software testing. In: SOMMERVILLE, I. (Ed.). **Software Engineering**. Boston, Massachusetts, EUA: Pearson Addison-Wesley, 2016. cap. 8, p. 144–163. Citado na página 55.

SOSKA, A.; MOTTOK, J.; WOLFF, C. An experimental card game for software testing: Development, design and evaluation of a physical card game to deepen the knowledge of students in academic software testing education. In: **7th IEEE Global Engineering Education Conference**. Abu Dhabi, United Arab Emirates: [s.n.], 2016. p. 576–584. Citado nas páginas 26, 27, 45 e 135.

SOUZA, D. M. **Subsídios à integração de ferramentas de avaliação automática e sistemas de gerenciamento de aprendizagem**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2017. Citado nas páginas 129, 130 e 137.

SOUZA, D. M.; COSTA, S. L. da; FILHO, N. F. D.; BARBOSA, E. F. Um estudo experimental do ambiente prog-test no ensino de programação. In: **X Workshop Latinoamericano Ingeniería de Software Experimental**. Montevideo, Uruguay: [s.n.], 2013. p. 75–88. Citado na página 26.

SOUZA, S. R. S. **Validação de Especificações de Sistemas Reativos : definição e análise de critérios de teste**. Tese (Doutorado) — Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2000. Citado nas páginas 41 e 44.

SOUZA, S. R. S.; SOUZA, P. S. L.; BRITO, M. A. S.; SIMãO, A. S.; ZALUSKA, E. J. Empirical evaluation of a new composite approach to the coverage criteria and reachability testing of concurrent programs. **Software Testing, Verification & Reliability**, v. 25, n. 3, p. 310–332, 2015. Citado na página 40.

SYAM, M. I. Possibility of applying flipping classroom method in mathematics classes in foundation program at qatar university. In: **1st International Conference on Social Sciences and Humanities**. Istanbul, Turkey: [s.n.], 2014. p. 180–187. Citado na página 50.

TAO, S. Y.; HUANG, Y. H.; TSAI, M. J. Applying the flipped classroom with game-based learning in elementary school students' english learning. In: **5th International Conference on Educational Innovation through Technology**. Tainan, Taiwan: [s.n.], 2016. p. 59–63. Citado nas páginas 28 e 68.

TAROUCO, L. M. R.; SCHMITT, M. A. R. Adaptação de metadados para repositórios de objetos de aprendizagem. **RENOTE - Revista Novas Tecnologias na Educação**, v. 8, n. 2, 2010. Citado na página 162.

TEIXEIRA, G. P. **Flipped Classroom: Um contributo para a aprendizagem da lírica camoniana**. Dissertação (Mestrado) — Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa, Lisboa, Portugal, 2013. Citado nas páginas 48 e 152.

THE INSTITUTE OF ELECTRONICAL AND ELECTRONICS ENGINEERS. **IEEE Standard Glossary of Software Engineering Terminology**. [S.l.], 1990. Citado na página 38.

THIRY, M.; ZOUCAS, A.; SILVA, A. C. Empirical study upon software testing learning with support from educational game. In: **XXIII International Conference on Software Engineering and Knowledge Engineering**. Miami Beach, USA: [s.n.], 2011. p. 481–484. Citado na página 155.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. **Introdução à engenharia de software experimental**. Rio de Janeiro, 2002. Disponível em: <http://cultura.ufpa.br/cdesouza/teaching/methods/6-ES-Experimental.pdf>. Acesso em: 07 dez. 2017. Citado nas páginas 64 e 66.

VALENZUELA, G. M.; JUNIOR, O. H.; SILVA, W. A.; ARAUJO, M. T.; MORAES, D. K.; FREIRE, A. A. C.; MORAES, M. D. S. Moodle auxiliando progressão parcial em escola pública. In: **XVIII Conferência Internacional sobre Informática na Educação**. Porto Alegre, Rio Grande do Sul, Brasil: [s.n.], 2013. p. 531–534. Citado na página 47.

VALLE, P. H.; BARBOSA, E. F.; MALDONADO, J. C. Um mapeamento sistemático sobre ensino de teste de software. In: **XXVI Simpósio Brasileiro de Informática na Educação**. Maceió, Alagoas, Brasil: [s.n.], 2015. p. 71–80. Citado nas páginas 32, 45, 155, 156 e 161.

VALLE, P. H. D. **Jogos educacionais: uma contribuição para o ensino de teste de software**. Dissertação (Mestrado) — Instituto de CIências MAtemáticas e de Computação - Universidade de São Paulo, São Carlos, 2017. Citado nas páginas 26 e 155.

VALLE, P. H. D.; BARBOSA, E. F.; MALDONADO, J. C. Cs curricula of the most relevant universities in brazil and abroad: Perspective of software testing education. In: **XVII International Symposium on Computers in Education**. Setubal, Portugal: [s.n.], 2015. p. 62–68. Citado nas páginas 32 e 147.

VICARI, R. M.; RIBEIRO, A.; SILVA, J. M. C. da; SANTOS, E. R.; PRIMO, T.; BEZ, M. Brazilian proposal for agent-based learning objects metadata standard - obaa. In: SÁNCHEZ-ALONSO, S.; ATHANASIADIS, I. N. (Ed.). **Metadata and Semantic Research**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 300–311. Citado nas páginas 174 e 175.

VINCENZI, A. M. R. **Orientação a objeto: Definição e análise de recursos de teste e validação**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2004. Citado na página 156.

WALLACE, R. S. The anatomy of a.l.i.c.e. In: EPSTEIN, R.; ROBERTS, G.; BEBER, G. (Ed.). **Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer**. Dordrecht: Springer Netherlands, 2009. cap. 13, p. 181–210. Citado nas páginas 57, 58, 212, 215, 221 e 224.

WEINTRAUB, J. **History of the PC Therapist**. 1986. Disponível em: <http://www.loebner.net/Prizef/weintraub-bio.html>. Acesso em: 22/02/2018. Citado na página 56.

WIGGINS, J. B.; GRAFSGAARD, J. F.; BOYER, K. E.; WIEBE, E. N.; LESTER, J. C. Do you think you can? the influence of student self-efficacy on the effectiveness of tutorial dialogue for computer science. **International Journal of Artificial Intelligence in Education**, v. 27, n. 1, p. 130–153, 2017. Citado na página 52.

WOHLIN, C.; RUNESON, P.; HST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLN, A. **Experimentation in Software Engineering**. New York City, USA: Springer Publishing Company, Incorporated, 2012. Citado nas páginas 28, 33, 61, 64, 89 e 92.

WOOD, H. Experiences and reflections of flipping the classroom. In: **2nd Flexible Futures**. [S.l.: s.n.], 2016. p. 18–26. Citado na página 50.

WOUDENBERG, A. van. **A Chatbot Dialogue Manager, Chatbots and Dialogue Systems: A Hybrid Approach**. Dissertação (Mestrado) — Faculty of Management,

Science and Technology - Open University of the Netherlands, Heerlen, Netherlands, 2014. Citado na página 51.

ZHAMANOV, A.; SAKHIYEVA, Z. Implementing flipped classroom and gamification teaching methods into computer networks subject, by using cisco networking academy. In: **12th International Conference on Electronics Computer and Computation**. Almaty, Kazakhstan: [s.n.], 2015. p. 1–4. Citado na página 49.

APÊNDICE

A

MAPEAMENTO SISTEMÁTICO SOBRE AGENTES CONVERSACIONAIS PARA EDUCAÇÃO

Research on Conversational Agents in Education: an overview of studies conducted in the last decade

Leo Natan PASCHOAL[1],
Aliane Loureiro KRASSMANN[2], Felipe Becker NUNES[2],
Simone do Rocio Senger de SOUZA[1]
[1]Institute of Mathematics and Computer Science, University of São Paulo
São Carlos/ SP, Brazil
[1]Interdisciplinary Center for New Technologies in Education, Federal University of Rio Grande do Sul
Porto Alegre/ RS, Brazil

Abstract: Conversational agents can be used in the educational area as a teaching support mechanism and can assist the student in resolving doubts. Although there are many studies that recognize the importance of conversational agents, they are still underexploited. Therefore, it is necessary to outline the research opportunities, the existing limitations and the main possibilities that these systems can offer. This paper contributes presenting a research mapping looking for identification, analyses and classification of primary studies in educational conversational agents. The results provide a characterization of the development technologies and models, implementation aspects, areas and educational levels where they have been used, and how they are evaluated. Conversational agent is a field of interest, however, some issues remain unexplored, for instance, the proposition of a development process. The results have pointed a range of possibilities for new studies to be developed.

Keywords: conversational agent, chatterbot, chatbot, conversational systems, systematic mapping.

1. Introduction

In the last decade, the interaction with digital systems has increasingly been transformed from scrolling, sliding, or button clicks to natural language text strings (Følstad and Brandtzæg, 2017), that is, interfaces based on natural language (Masche and Le, 2017). With the improvement of data-mining, machine-learning techniques and robust linguistic annotations/processing tools standards, like eXtensible Markup Language (XML) and their applications, conversational agents are becoming more practical in daily life applications (Ghose and Barua, 2013).

Radziwill and Benton (2017) estimate that because of their flexibility and ease of use, conversational agents can become an universal user interface, a democratization of the versatile command line. This happens because users are comfortable on exchanging information through short text messages among friends and colleagues (Zumstein and Hundertmark, 2017), a trend also prevalent among students (Crown et al., 2010). In this scenario, Kane D. (2016) and Schmidt (2018) noted that the conversational agents are a natural fit for providing reference and instructional assistance and for this reason they are increasingly being used in education, allowing the supply of individualized and personalized tutoring.

Individualized tutoring (a student working with an instructor) has been used for a long time as an effective instructional procedure, allowing to adjust the rhythm of instruction to the needs and abilities of each student, and in consequence resulting in good learning achievements,

which was nominated 2 Sigma Problem (Bloom, 1984). One of the reasons of this outcome, as highlighted by Fletcher (2001), is because most classrooms contain students who, at one end of the spectrum, are bored and on the other hand are overloaded and lost.

However, as even Fletcher (2001) emphasizes, unless educational funding policies change dramatically, we cannot afford a single teacher for each student. Bloom himself (1984, p. 44) recommends that research should focus on "finding ways to accomplish this on more practical and realistic terms than private lessons, which is too expensive for most societies to support on a large scale".

In this sense, conversational agents become a cost-efficient, feasible and user-friendly alternative. It has already been identified that Intelligent Tutoring Systems (ITS) with sophisticated pedagogical tactics, but without natural language dialogue, produces an effect of only about 1 Sigma (Corbett, 2001). Winkler and Söllner (2018) discovered that students appreciate the use of conversational agents because of a more error-friendly learning environment, compared to conversations with human authorities.

Despite all these advantages, conversational agents are still in the early stages of their expansion. According to Chatbot's Magazine (2017), 2018 promises to be a year in which conversational agents will spread to more fields such as education. We believe that the slowness is this process due the existent drawbacks, some challenges that need to be overcome in order to promote the use of conversational agents in education, from technical and pedagogical points of view, as the choice of technologies and the evaluation methods.

In order to contribute with this discussion, this paper presents a systematic mapping study conducted for the identification of the scenarios of development and application of conversational agents in context of education, looking for answer the following Research Question (RQ): ***how are conversational agents being developed, applied and evaluated in education context?***

This paper brings a panoramic view of the use of conversational agents in education area, with the analysis and summary of the covered knowledge areas. In order to contribute with the development of conversational agents for education, we present the technical and educational requirements involved in the development of these approaches (platforms, multimedia resources, educational theories and graphic representations); and how the conversational agent approaches are being applied and evaluated, taking in to account factors analyzed; types of studies; target audience; data collection instruments and data analysis methods. The results consider an overview of ten years of scientific contributions in this context.

2. Background

Conversational agents are commonly applied in e-commerce, information retrieval, helpdesk tools, customer support and automated answering systems. In these areas, the economy is estimated to an average $ 0.70 per interaction compared to traditional support conversations (Juniper Research, 2017). According to Business Insider (2017), the potential annual salary

savings created by conversational agents is estimated at $23 billion. Using conversational agents, consumers and businesses can communicate 24 hours a day, 7 days a week, regardless of office hours or work hours (Zumstein and Hundertmark, 2017), which increase user satisfaction by making response time immediate. Although one of the most prolific, well documented and useful applications relates to its use in education (de Gasperis et al., 2013), all this cost efficient potential yet has to be explored in this field.

The architecture of a conversational agent integrates a language model and computational algorithms to emulate communication between a human user and a computer using natural language (AbuShawar and Atwell, 2016). The technology had its begin at the Massachusetts Institute of Technology (MIT), with the Eliza conversational agent, designed in the late 1960s by Weizenbaum (1966) as a fictitious psychotherapist engaged in a conversation with users using a word matching technique, which the idea is based on finding the best and longest matching pattern. Since then, numerous solutions have emerged, using Artificial Intelligence (AI) and machine learning techniques, such as Word2vec, Markov Chains, Neural Networks and Façade (Mateas and Stern, 2004); and models that also use pairs of patterns as Eliza, such as Cleverscript (Cleverbot, 2016), ChatScript (Wilcox, 2011) and AIML (Wallace, 1995).

Conversational agents can assume different terminologies, such as chatbots, chatterbots or digital assistants. We must distinguish among technological terms that have other goals and functionalities, as dialogue systems or spoken dialogue systems, that are usually a software with preprogrammed “dialogs” (scripted), in order to present an instructional content to the user in a more interactive way, but it does not have the property of real “dialogue”, i.e., understanding natural language. Perez-Marin and Pascual-Nieto (2011), on the other hand, mention that the main difference is associated with the architecture of each software, since a dialog system includes a module for managing the dialog which in turn contains artificial Intelligence features. This module is used to retrieve possible actions to be performed, responses via real reasoning about known facts of the domain or about the context of the dialogue.

Also, unlike goal-oriented dialogue systems (Wen et al., 2016; Lee et al., 2009), a conversational agent is aimed to chat with human users on any subject domain of daily lives (Serban et al., 2016; Shang et al., 2015). Jurafsky & Martin (2018) clarifies that task-oriented dialog agents are designed for a particular task, and set up to have short conversations to get information from the user to help complete the task. These include the digital assistants that are now on every cell phone or on home controllers (Siri, Cortana, Alexa, Google Now/Home, etc.) whose dialog agents can give travel directions, control home appliances, find restaurants, or help make phone calls or send texts. On the other hand, chatbots are systems designed for extended conversations, set up to mimic the unstructured conversational or ‘chats’ characteristic of human-human interaction, rather than focused on a particular task like booking plane flights.

In educational area, conversational agents can help in overcoming the difficulties inherent to the simultaneous attendance of a larger number of students, as in online courses, providing individualized treatment. Also, they can conveniently retrieve information without the student

having to look or browse several web pages to fetch answers of frequently asked questions (Ghose and Barua, 2013). Research has already identified their potential in providing greater safety and motivation for students. According to Graesser et al., (2014), learning is facilitated compared to students who read textbooks and engage in non-interactive learning environments, just as human tutors are more effective than reading textbooks.

Keystone research in this field by Mayer and colleagues (e.g., Mayer et al., 2003; Moreno et al., 2001) have demonstrated that the action of an intelligent agent was more effective when explanations to the student were in the form of speech rather than only on-screen text. While the audio/visual support that technology provides is important for all education, the possibility of conversational interaction with an intelligent agent is at the heart of technology's potential contribution learning (Fryer et al., 2017).

Considering this potential, this research presents an overview on the development and application of conversational agents in the formal education context, through established peer-reviewed studies from the last ten years, aiming to understand how it is possible to take advantage and to extend its educational use. The formal education scope was defined aiming to comprise only studies developed to be applied and replied in schools, universities and formal educational institutions, disregarding the ones with a more wider or abstract focus, like lifelong learning or corporate trainings.

3. Related Works

Growing advance in the development of conversational agents in the educational field has been explored by different literature reviews that have been developed over the last few years. In a majority consensus from the related studies (Mdehn and Mulken, 2000; Rickel, 2001; Kasap and Magnenat-Thalmann, 2008; Winkler and Söllner, 2018), it is possible to verify that the conversational agents applied in the educational scope are still in a growing emerging, in spite of the great amount of existing works. Diversified works are developed in different types of educational approaches, supporting tools and employing different types of technologies. The results obtained with the systems' applications have a positivist scenario, in which the works have demonstrated the pedagogical potential that the use of this type of resource can have in the learning process of the students.

However, studies show that there are still many challenges to be overcome in the development of conversational agents for education because of the constant evolution of technology resources and the individual needs of students, which requires constant updating of the approaches developed. The systematic literature reviews carried out presents a consensus: the process of constructing a conversational agent and its knowledge base for the educational scope is complex and costly, since there are many variables to be treated, as the knowledge to be used, support technologies and educational theories that will base communication and interaction with students (Winkler and Söllner, 2018).

In the work of Schroeder and Adesope (2014), the affective implications in pedagogical agents is analyzed by means of a systematic review. The results suggest that learners may prefer

pedagogical agents compared to agent-free control conditions, and pedagogical agents can provide motivational benefits for some populations under certain conditions. However, it is unclear whether pedagogical agents impose external cognitive load. The work concludes that the graphical representation of the agent may not be necessary for the student to receive the affective benefits of the interaction.

Schroeder et al. (2013) conducted a systematic literature review focusing on pedagogical agents towards education, reviewing 43 studies involving 3,088 participants. Analysis of the results indicated that pedagogical agents produced a small but significant effect on learning. The overall mean effect was moderated by the contextual and methodological characteristics of the studies. The results showed that the use of pedagogical agents was more beneficial for elementary and middle school students than high school students. Pedagogical agents who communicated with students using text on the screen facilitated learning more effectively than the agents who communicated using narration.

Further systematic reviews carried out research with a more restrictive scope (e.g., Ontologies, Foreign Learning). Work developed by Pereira Júnior et al. (2017) aimed to investigate how ontologies can contribute to the development of conversational agents in the learning context. A total of 10 primary studies were reviewed after an initial selection of 113 papers. Results indicate that the use of Ontologies in the conversational agents is promising, since they allow actions in different domains, with different methods of interaction, inclusion of aspects of affectivity, and the need to meet non functional requirements such as usability and performance. Despite the benefits found, it is known that the creation of ontologies is costly.

Along to this context, there is the use of pedagogical agents to improve the English language learning, which was demonstrated in the systematic review of Murtiningsih (2011). The author understands that because English is a language that is not used in daily communication and the variety of students' language competence, the use of first language in a foreign language classroom has also become another point to encourage the students' foreign language acquisition. He concluded that this type of approach has gained positive space in the academic world.

Based on the analysis carried out in the previous work, it became possible to create a consistent know-how for the construction of this research. One of the missing gaps that have been observed is the scarcity of systematic literature reviews covering all the process of conversational agents in education, since the elaboration, design and theories to be associated, to the development, construction, implementation, application and evaluation of systems and approaches.

In order to provide a complete view that ranges from the beginning of the construction of the research prototype until its final phase of application, this research was carried out, centered on conducting a systematic mapping study of conversational agents in the formal educational field, with the purpose of providing subsidies to other researchers in this area.

4. Systematic Mapping Plan

The conduction of this Systematic Mapping Study (SMS) is based on a predefined protocol that considers the guidelines defined by Petersen et al. (2008) and Petersen et al. (2015). Therefore, the SMS was divided into five stages, being: (i) definition of research questions, (ii) search for primary studies, (iii) screening of studies for inclusion and exclusion, (iv) keywording of abstracts, and (v) data extraction and mapping of studies. It was defined that the mapping would contemplate a delimited period, between 2007 and 2017. This period was established because we are interesting in works that consider more recent technologies.

4.1. Definition of research questions

The GQM (Goal/Question/Metric) approach (Basili, 1992) was adopted and research issues were defined from objectives, which are achieved by means of a set of metrics that will enable them to answer the questions. In this study the goal was to provide a state of the art on conversational agents developed to support formal education practices, focused on: (i) what are the main areas of knowledge in which the investigation of conversational agents has been applied? (ii) how conversational agents have been developed? and, (iii) how conversational agents have been evaluated?. Based on this goal, three research questions, eleven subquestions, and related metrics were established, as presented in Table 1.

Table 1
Research questions and metrics following the GQM model

Research questions		Metrics
RQ_1	**Which educational area the conversational agents have been applied?**	**M_1: Body of evidence constructed from the subquestions of RQ1**
	$RQ_{1.1}$: What are the main educational levels of regular education in which the application of conversational agents has been investigated?	$M_{1.1}$: Educational levels of formal education that have been identified in primary studies.
RQ_2	**How the conversational agents used in educational area have been developed?**	**M_2: Body of evidence constructed from the sub questions of RQ_2.**
	$RQ_{2.1}$: What are the logics of systems, techniques or approaches have been used for implementation?	$M_{2.1}$: Logics of systems that have been used (e.g., Pattern Matching, Ontologies, Markov Chain Models).
	$RQ_{2.2}$: Which programming languages have been used for implementation?	$M_{2.2}$: Programming languages that have been used to implement the system.
	$RQ_{2.3}$: What are the communication strategies that enable natural language interaction with the user that has been used in the implementation of conversational agent?	$M_{2.3}$: Multimedia resources have been used by conversational agents to communicate with the student (e.g., text, audio, image).
	$RQ_{2.4}$: Which software process models have been used in this development?	$M_{2.4}$: Software process models that have been used.
	$RQ_{2.5}$: Which learning theories or educational principles have been used to subsidize this development?	$M_{2.5}$: Learning theories and educational principles that have been used.
	$RQ_{2.6}$: How have the conversational agents been represented?	$M_{2.6}$: Graphical appearance of the conversational agents that have been identified in primary studies.
RQ_3	**How the conversational agents used in educational area have been evaluated?**	**M_3: Body of evidence constructed from the sub questions of RQ_3.**

$RQ_{3.1}$: Which aspects/variables have been selected for evaluation?	$M_{3.1}$: Set of aspects/response variable conversational agent that have been identified for evaluation in the primary studies (e.g., motivation, usability).
$RQ_{3.2}$: What empirical method have been used?	$M_{3.2}$: Types of empirical evaluation approaches that have been identified in primary studies (e.g., case study, experiment, literature survey).
$RQ_{3.3}$: Which data collection instruments have been used?	$M_{3.3}$: Set of instruments have been identified in primary studies (e.g., questionnaire, interaction recording).
$RQ_{3.4}$: Which data analysis methods have been used?	$M_{3.4}$: Statistical methods used in the evaluation analysis have been identified in primary studies (inferential statistics or descriptive statistics).

4.2. Search procedures

After the definition of the research questions (RQ), the search string was constructed, so that the searches could be carried out in the most relevant databases that index primary studies about educational conversational agents. Six databases were selected and presented in Table 2. According to Dyba et al. (2007) and Kitchenham and Charters (2007), these databases are the most relevant sources in the Computing area. Additionally, the Engineering Village database was adopted based on the experience reported by Petersen et al. (2015).

Table 2
Databases used in the search

Database	Source
ACM Digital Library	http://portal.acm.org
Elsevier (Science Direct)	http://www.sciencedirect.com
Engineering Village	https://www.engineeringvillage.com/search
IEEE Xplore	http://ieeexplore.ieee.org
Scopus	http://www.scopus.com
Web of Science	https://webofknowledge.com

The search string was defined so that it could contemplate a big range of variations of the terms "conversational agent" and "education" existing in the literature. Therefore, the search string was built around the union of these keywords, with their spelling variants and associated terms. Figure 1 illustrates the variations of the terms that were used to define the search string.

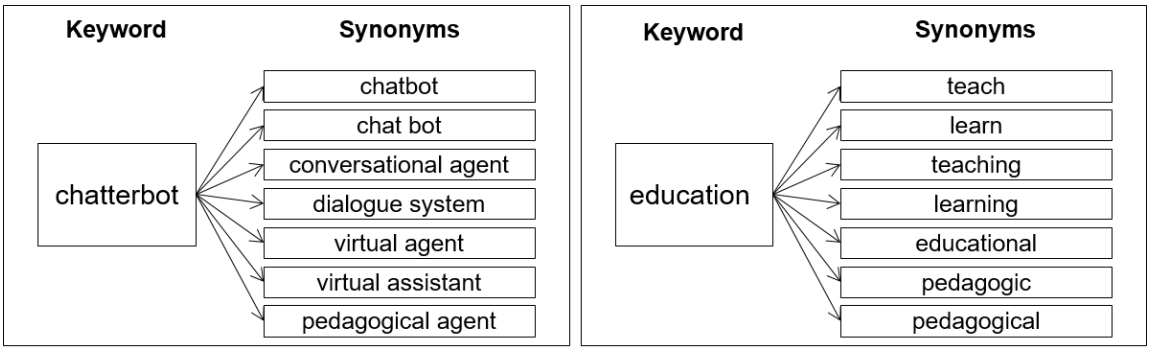


Fig. 1. Terms that form the search string

The search string was created based on terms of Figure 1: ((chatterbot OR chatbot OR "chat bot" OR "conversational agent" OR "dialogue system" OR "virtual agent" OR "virtual assistant" OR "pedagogical agent") AND (education OR pedagogic OR learn OR teach OR educational OR pedagogical OR learning OR teaching)). The logical operator “OR” was used to indicate that the alternate terms were acceptable, and the operator “AND” was used to connect the two sides of the string.

In order to select the studies, factors that identify the relevance of the primary studies in solving the research questions were defined and inclusion and exclusion criteria were defined (Table 3). These criteria were used as a parameter to assist the authors during the analysis of the primary studies, making the analysis more reliable and accurate

Table 3
Inclusion and Exclusion Criteria

Inclusion Criteria (IC)	**Exclusion Criteria (EC)**
IC-1: The primary study presents at least one conversational agent to be used in the context of formal education.	EC-1: The study is a technical report, a document that is available in the summary format, is a presentation, a call for paper, a summary of a conference.
IC-2: The primary study discusses about at least one conversational agent that has the ability to simulate a conversation with the user via natural language.	EC-2: The primary study is written in a language other than English.
IC-3: The primary study describes the process of development, the application and evaluation of at least one conversational agent.	EC-3: The primary study does not provide an identification of the authors in the body of the text.
	EC -4: The full text of the primary study is not available.
	EC -5: The primary study does not address the actual development, application and evaluation of a conversational agent in the educational sphere, that is, it has only one of the terms used in the search string in the body of the article, or presents only theoretical aspects or proposals/prototypes under construction.
	EC-6: The primary study presents a conversational agent that has the ability to communicate with the user using different natural language resource.

By focusing on formal learning, we have considered approaches applied in education normally delivered by trained teachers in a systematic intentional way within a school, higher education or university. That is, other contexts of education as training, short courses, etc., were disregarded. In this process, we have also discarded publications that are not directly related to conversational agents, that is, related to goal-oriented dialogue systems as stated on the Background section.

4.3. Review Execution

The process of selection of the primary studies was carried out following some phases (Figure 2):

Phase 1. First, a search was conducted in each database and the initial search resulted in a total of 5.075 primary studies. The metadata of these primary studies containing the title, summary, article URL, keywords, and number DOI (Digital Object Identifier), were exported from these bases, in a BIB file format. Subsequently, the metadata was imported into the Parsif[1] tool, which supports the automatization of some activities while conducting systematic mappings and/or reviews of literature.

Phase 2. In this phase the duplicated primary studies were removed, with the support of Parsif tool. After that, it was remained 2.619 primary studies

Phase 3. The remaining 2.619 primary studies were randomly distributed among the authors. In this initial analysis, the authors read the title, abstracts and keywords and applied the inclusion and exclusion criteria (first round). In the end of this phase, the number of primary studies was reduced to 548.

Phase 4. The authors carefully read each paper remaining after Phase 3 and kept only those relevant to the research questions. As a result of this phase, 101 studies[2] were selected.

Phase 5. The 101 selected studies were analyzed and data were extracted to answers the research questions.

[1] More information can be obtained at: <https://parsif.al/>.
[2] The list of primary studies that have been selected is available at: <http://bit.ly/2SzmrvK>

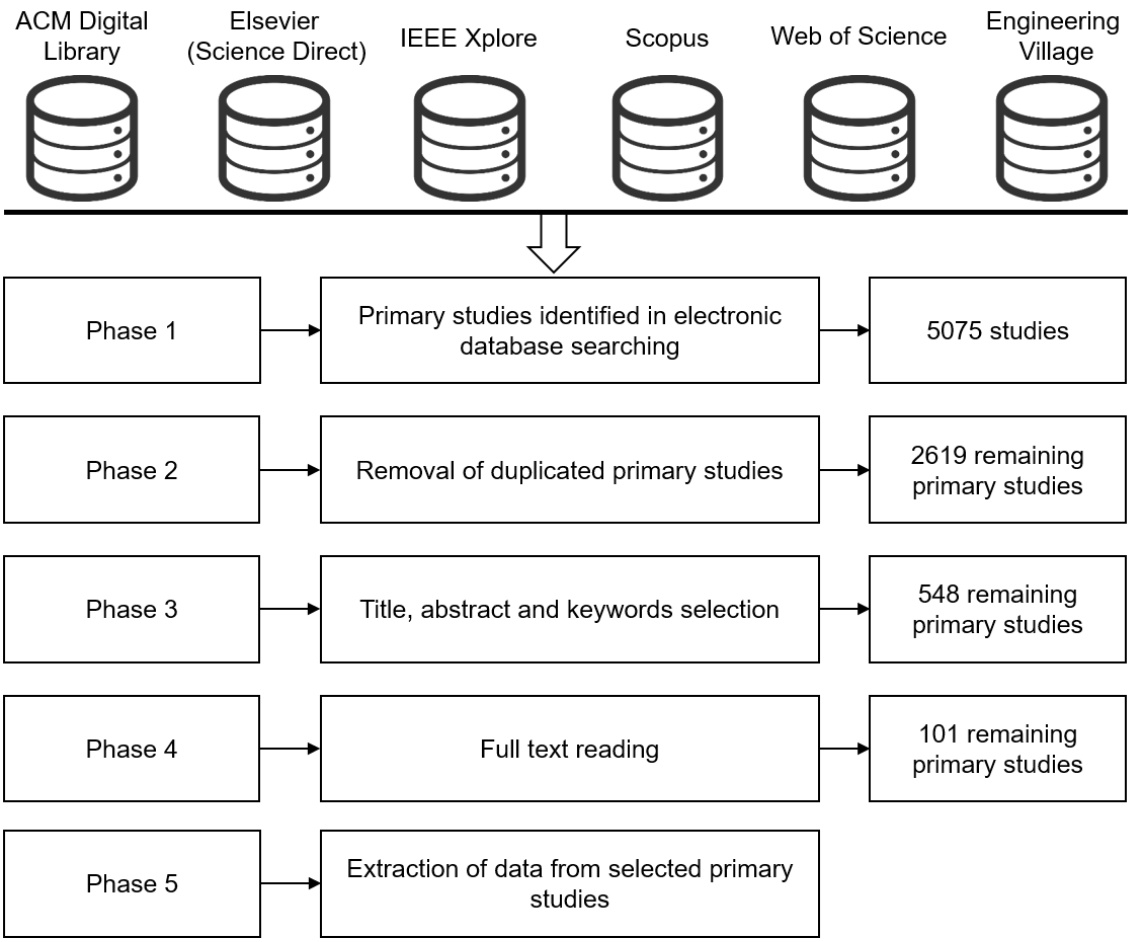


Fig. 2. Study selection procedure

5. Reporting the Mapping

In this section, we discuss the results of the SLR, with an overview of the primary studies that were extracted. To present our findings, we have structured this section by research question. Given this, initially, in Section 5.1 we present the distribution of selected studies over the years and the publishing vehicles. Subsequently, in Section 5.2 we present the identification of the educational areas that are being supported by the conversational agents and reveal the relationship between them and the levels of education. In Section 5.3.we demonstrate how the conversational agents are being developed, indicating elements that involve encoding and the final product. Finally, Section 5.4 describes the results related to the evaluation of conversational agents.

5.1. Overview of the Selected Studies

Figure 3 presents the temporal evolution of publications, pointing out the number of primary studies per year. We can observe that the amount of studies per year has been increasing since the year 2007, despite some casualties at certain years (as 2009 and 2015). We believe this

shows that the subject is becoming more and more relevant within the community, as there is an increase in interest for the scientific community in the area. We also managed to realize that there is a concentration of publications between the periods from 2010 to 2014. This concentration returned in the year 2016 and settled since then.

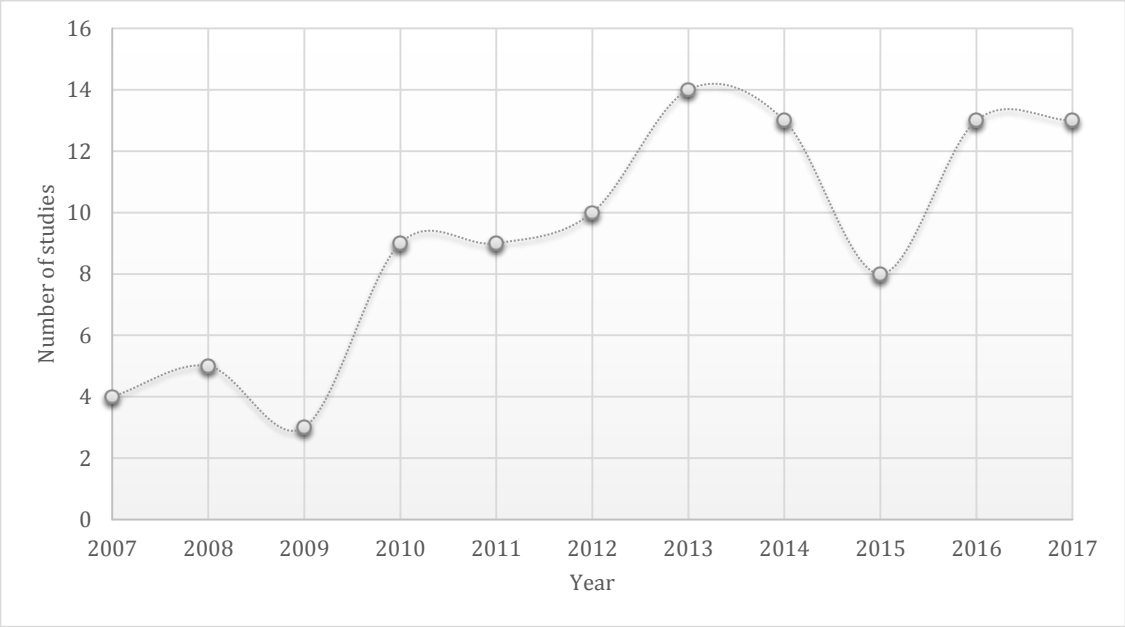


Fig. 3. Number of primary studies published per year

Primary studies were published in different scientific dissemination vehicles. We classify these publications into three categories, considering the type of vehicle in which the primary studies were published: scientific journals, conference proceedings and book chapters. In Figure 4 is illustrated this distribution, showing that the most of the primary studies were published in Conferences (59 primary studies); 36 primary studies in Journals and a smaller number (6) in in Book chapter.

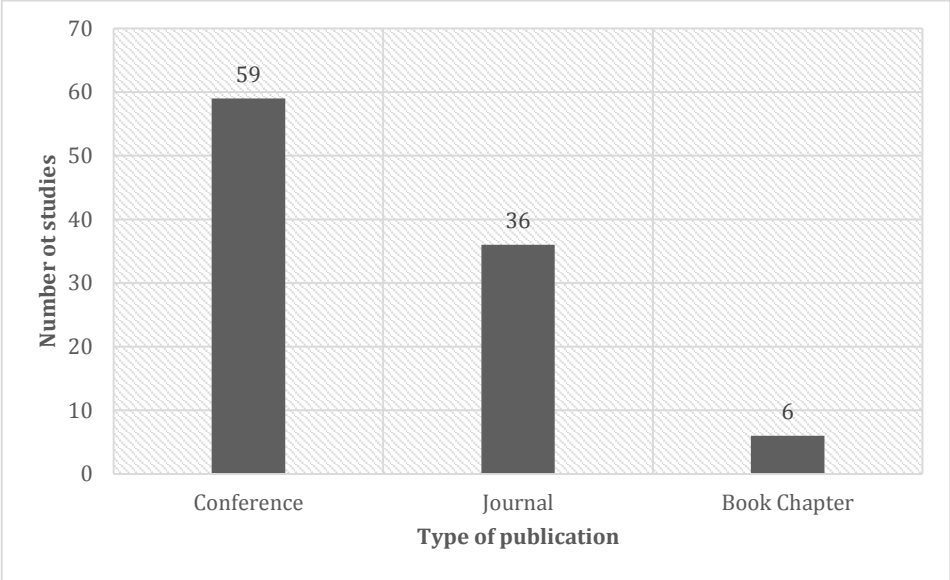


Fig. 4. Primary studies organized by publication type

In addition this analysis, we also identified the scientific vehicles that disseminated the primary studies (Table 4). This mapping study covered 26 different journals, 40 conferences, and 5 book chapters. We can observe that studies have been published in channels of the area of Artificial Intelligence, Human-Computer Interaction, and, especially, in channels that publish interdisciplinary studies of Educational Technology. With respect to the venues in which studies were published, we cannot observe a single location that concentrates the studies.

Table 4
Channels for the publication of primary studies

Channels for publication	**Number of studies**	**Publication type**
Advances in Intelligent and Soft Computing	1	Journal
Annual Conference of the North American Chapter of the Association for Computational Linguistics	1	Conference
Annual Conference on Innovation and Technology in Computer Science Education	1	Conference
Annual Meeting of the Special Interest Group on Discourse and Dialogue	1	Conference
Applied Artificial Intelligence	1	Journal
Artificial intelligence in education: Building technology rich learning contexts that work	1	Book Chapter
Australasian Society for Computers in Learning in Tertiary Education Conference	1	Conference
Behaviour & Information Technology	1	Journal
British Journal of Educational Technology	1	Journal
Computer Assisted Language Learning	1	Journal
Computers & Education	5	Journal
Computers in Human Behavior	1	Journal
Conference on Human Factors in Computing Systems	2	Conference
Conference on User Modeling, Adaptation and Personalization	3	Conference
Education and Information Technologies	1	Journal
Educational Media International	1	Journal
Educational Technology and Society	1	Journal

Educational Technology Research and Development	1	Journal
E-Learning Paradigms and Applications: Agent-based Approach	1	Book Chapter
European Conference on Technology Enhanced Learning	2	Conference
Hawaii International Conference on System Science	1	Conference
IADIS International Conference Interfaces and Human Computer Interaction	1	Conference
IEEE Frontiers in Education Conference	1	Conference
IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning	1	Conference
IEEE Spoken Language Technology Workshop	1	Conference
IEEE Transactions on Learning Technologies	4	Journal
IEICE Transactions on Information and Systems	1	Journal
Interactive Learning Environments	1	Journal
IEEE International Conference on Advanced Learning Technologies	3	Conference
International Conference on Advances in Web-Based Learning	1	Conference
International Conference on Artificial Intelligence in Education	3	Conference
International Conference on Autonomous Agents and Multiagent Systems	3	Conference
International Conference on Computer Graphics Theory and Applications	1	Conference
International Conference on Computer Supported Collaborative Learning	1	Conference
International Conference on Computers in Education	1	Conference
International Conference on Digital Information Management	1	Conference
International Conference on Human System Interactions	1	Conference
International Conference on Human-Agent Interaction	1	Conference
International Conference on Informatics, Electronics and Vision	1	Conference
International Conference on Innovative Techniques and Applications of Artificial Intelligence	1	Conference
International Conference on Integration of Knowledge Intensive Multi-Agent Systems	1	Conference
International Conference on Intelligent Networking and Collaborative Systems	1	Conference
International Conference on Intelligent Technologies for Interactive Entertainment	1	Conference
International Conference on Intelligent Tutoring Systems	4	Conference
International Conference on Intelligent Virtual Agents	6	Conference
International Conference on Interdisciplinary Research on Technology, Education and Communication	1	Conference
International Conference on Natural Language Processing and Knowledge Engineering	1	Conference
International Conference on Technological Ecosystems for Enhancing Multiculturality	1	Conference
International Conference on User Science and Engineering	1	Conference
International Conference on Distributed Computing and Artificial Intelligence	1	Conference
International Conference on Intelligent Tutoring Systems	1	Conference
International Journal of Advanced Computer Science and Applications	2	Journal
International Journal of Artificial Intelligence in Education	2	Journal
International Journal of Human-Computer Studies	1	Journal
International Symposium on Computers in Education	2	Conference
International Technology, Education and Development Conference	1	Conference
Journal of Ambient Intelligence and Smart Environments	1	Journal
Journal of Educational Computing Research	1	Journal
Journal of Multimedia	1	Journal
Journal of Social Work Education	1	Journal
KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications	1	Conference
Knowledge-Based Systems	2	Journal
Lecture Notes in Computer Science	2	Book Chapter

Lecture Notes in Engineering and Computer Science	1	Book Chapter
Panhellenic Conference on Informatics	1	Conference
Patient Education and Counseling	1	Journal
Procedia Computer Science	1	Journal
Recent Researches in Engineering Education and Software Engineering	1	Book Chapter
SIGCSE technical symposium on Computer science education	1	Conference
Simulation in Healthcare	1	Journal
The Journal of Personalization Research	1	Journal

In order to answer the research questions, we conducted a deeper analysis of the 101 primary studies that were extracted. These questions aim to highlight evidences about the state of the art in the context of conversational agents applied to formal education.

5.2. Which educational area the conversational agents have been applied?

The first issue that we investigated was the area of knowledge in which educational conversational agents have been investigated, and used as teaching support mechanism. This information is usefull for teachers or researchers from these areas because they can locate similar studies more easily, observe scientific gaps and use related studies to develop new scientific research.

Figure 5 shows that 29 studies did not mention the area of application of the conversational agents, or do not related the conversational agent to a specific domain (classified as “Others”). In these studies, the authors present generic conversational agents that can be used for formal education, without being a conversational agent intended to support the teaching of a specific content (e.g. a study seeks to investigate the effects of an agent's presence and different types of feedback on learning, motivation and cognitive load in a multimedia-learning environment (Lin et al. 2013). Fourteen studies focus on language teaching and learning, which we classified in the Linguistics area. These studies establish contributions to the study of languages, such as English, Arabic, and other languages, and they mention that the conversational agents encourage student training, turning easily the learning of new language. They enable the student to exercise the formulation of phrases and questions, know the meaning of words and exercise the spelling of terms.

We also found 13 primary studies with focus on Computer Science education, in which conversational agents are used to support teaching and learning practices in different areas of computing. These studies mention the importance of using a mechanism that supports natural language guidance in resolving students' doubts and difficulties while they learning themes as programming, database, and computer networks.

In a smaller proportion, we found nine studies about the teaching of medicine and nursing, which were classified in the area of Health, and eight studies focusing on Mathematics, with particular emphasis in calculus, linear algebra and analytical geometry. Others areas of application of the extracted studies can be viewed in Figure 5.

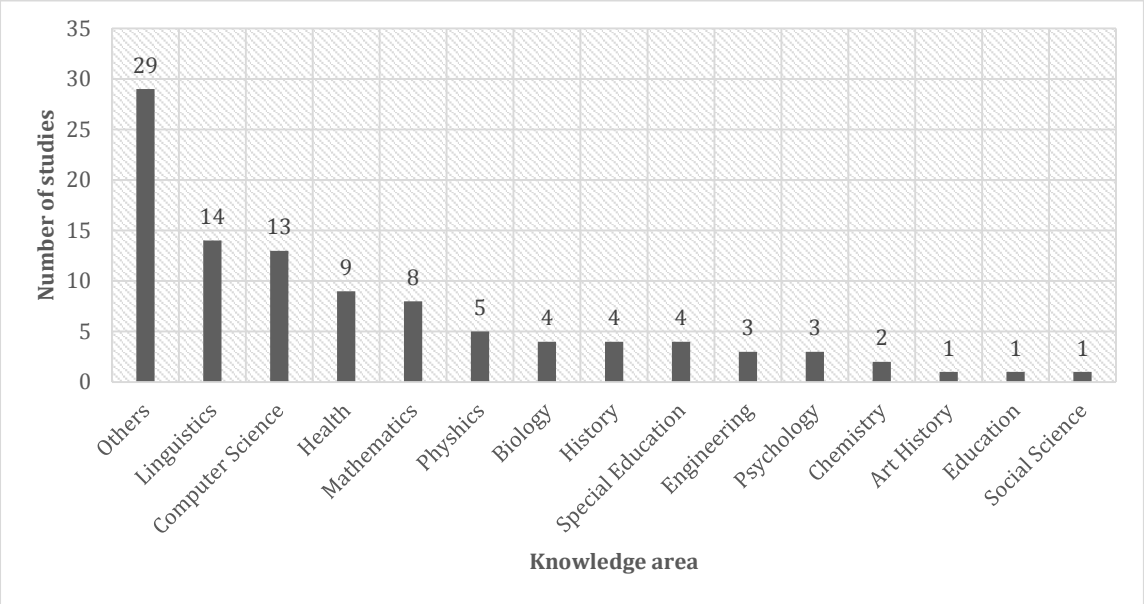


Fig. 5. Application domain of each primary study

In addition, we also locate the educational level that each conversational agent is destined, and relate them to the areas of knowledge. This is important because it shows that each conversational agent was developed to support the teaching of the content to a specific target audience. Thus, the educational level also characterizes the target audience. We carefully analyze the audience mentioned by each primary study, and classify these studies based on the guidelines of Organization of U.S. education3. Table 5 presents the distribution of primary studies by educational levels.

Table 5
Distribution of primary studies by educational levels

		Number of studies by knowledge area														
		Others	**Linguistics**	**Computer Science**	**Health**	**Mathematics**	**Physics**	**Biology**	**History**	**Special Education**	**Engineering**	**Psychology**	**Chemistry**	**Art History**	**Education**	**Social Science**
Number of studies by educational levels	**Undergraduate student**	10	2	5	2	—	2	—	2	—	3	3	1	—	—	—
	Graduate student	8	4	4	2	—	1	—	—	2	—	—	—	—	—	1
	First stage of basic education	4	2	—	2	1	—	2	1	—	—	—	—	—	—	—
	Second stage of basic education	1	1	1	1	2	—	2	1	1	—	—	—	—	—	—
	(Upper) Secondary education	1	—	2	—	2	2	—	—	—	—	—	1	—	—	—

[3] More information available at: <http://bit.ly/2LakRQQ>

Undergraduate student and Graduate student	3	—	—	—	—	—	—	—	—	—	—	—	—	1	—
Second stage of basic education and (Upper) Secondary education	—	1	—	—	2	—	—	—	—	—	—	—	—	—	—
Pre-primary education	1	—	—	—	—	—	—	—	1	—	—	—	—	—	—
Uninformed	1	1	—	1	—	—	—	—	—	—	—	—	—	—	—
(Upper) Secondary education and Graduate students	—	1	—	1	—	—	—	—	—	—	—	—	—	—	—
(Upper) Secondary education and Undergraduate student	—	—	1	—	—	—	—	—	—	—	—	—	—	—	—
Second stage of basic education and Undergraduate student	—	—	—	—	—	—	—	—	—	—	—	—	1	—	—
First stage of basic education, (Upper) Secondary education, and Undergraduate student	—	—	—	—	1	—	—	—	—	—	—	—	—	—	—
Undergraduate student and Second chance school	—	1	—	—	—	—	—	—	—	—	—	—	—	—	—

It is possible to observe in Table 5 that primary studies are concentrated mainly on the line that represents the educational level "Undergraduate Students", followed by the level "Graduate Students". Also, we can identify that the primary studies classified in the "Others" category, for the most part, although not specified the domain of application, specify the level of education to which the conversational agent is intended. In relation to the area of Linguistics, we can notice, for example, that a greater number of studies focus on the educational level "Graduate Students". It is believed that this is related to the fact that graduate programs require students the proficiency in a second and/or third language. This may also be related to the need for graduate programs students to study materials published in different languages.

In addition, by means of Table 5 we realize that the primary studies destined to Computer Science education do not only have as a target audience undergraduate students. Given the various incentives to incorporate computing in school curricula, there is a research strand that has been seeking to incorporate the teaching of computing, in particular of programming logic,

in basic education. This fact was highlighted in the SLR, in which we extracted studies about conversational agents to support computing teaching to students from basic education.

Regarding the Health area, we locate studies on education of students, both from higher and basic education. In higher education there are efforts to develop conversational agents aiming to support of training of medical students, in which, for example, they interact with a conversational agent that simulates a sick human, and they need to diagnose which possible disease the conversational agent has. In contrast, in the context of basic education, studies focus on health education, that is, conversational agents that seek to encourage students to practice good hygienic habits, disincentive drugs using, among others.

Different from what was noted by extracting information pertaining to the area of knowledge, in which it was observed 29 primary studies which do not specify the focus, we were able to identify 99 primary studies that specify to which educational level the conversational agent is intended. Only two studies do not mention it, one with focus on Health and another that had been classified in "Others" category. Thus, we identified that conversational agents developed for the different areas of knowledge mention, in general, the educational level. With this, it is possible to establish the target audience of the study.

5.3. How the conversational agents used in educational area have been developed?

Different types of techniques, methods, and frameworks are used during the development of conversational agents. The first aspect observed was the system logic, or inference machine, employed into conversational agents. This aspect represents the way that conversational agents manage to understand and interact with the user (Abdul-Kader; Woods, 2015).

In Table 6 is presented the system logic most cited in the primary studies. We have identified 27 different types of systems logic that are being used, which we categorize into: Bazzar agent framework; ccLDA (cross-collection Latent Dirichlet Allocation); Cleverbot; Created by the authors; Decision tree; Finite state machine; Fuzzy Logic; HDCM (Human-centered Distributed Conversational Modeling) and Case-based reasoning; Lexical network analysis; MASCARET meta-model; Naive Bayes; Neural Networks; Neural Networks and Bayesian models; OpenNLP; Parsing; Parsing and Ontologies; Pattern Matching and Concept Maps; Pattern Matching and FreeLing; Pattern Matching and others; Pattern Matching and Parsing; Pattern Matching and Rules created by the authors; Policy Iteration; RUANLP algorithm; and the techniques described in the Table 6. Some studies used only one of them during the implementation of the conversational agent. On the other hand, other studies mention the use of a mix of techniques during the implementation of the conversational agent. More than half of the studies (52) did not mention which logic was used the implement their system.

Table 6
Logic of systems used in the implementation of the conversational agents

Techniques	Number of primary studies

Uninformed	52
Pattern Matching	17
Ontologies	4
Bag of Words	3
Markov Chain Models	2

Pattern matching is the most used in the analysed papers, with 17 studies mentioning this technique Other studies also make use of Pattern Matching in conjunction with other logics and tools, such as Parsing, Concept Maps, the FreeLing[4] Natural Language analysis tool, among others. We also note that some studies are concentrating efforts on using advanced Artificial Intelligence techniques such as Fuzzy Logic, Naive Bayes, Neural Networks, Decision Trees, Markov Chain Models, but in a not very significant number (one study each). As a result, we found that the state of art in the development of conversational agents resemble the techniques used in international competitions such as the Loebner Prize Competition[5]. According to Abdul-Kader et al. (2015), since its emergence, the Pattern Matching remains as the most effective and most used method in the implementation of conversational agents.

Programming language. Similar as the previous analysis, we observed that 51 studies do not mention the programming language used (one less than the previous analysis),. This shows that although one study did not inform the technology, it informed the programming language used. The XML appears in 29 studies, which we believe is mostly related to the Pattern matching logic used in the implementation of conversational agents, alone or in association with another technique, which was shown in a good portion of the studies (23).

Figure 6 shows a bubble chart with the relationship among programming language and technology or system logic used in the conversational agent. We noticed that out of the 52 studies that did not mention system logic, 14 of them used XML. In this regard, we also found that the XML language is not used only in the encoding of conversational agents developed with Pattern matching technique. We realized that one study used XML in conjunction with Neural Networks, another study used XML in conjunction with Markov Chain models and another study used XML in conjunction with Fuzzy Logic.

[4] http://nlp.cs.upc.edu/freeling
[5] http://www.aisb.org.uk/events/loebner-prize

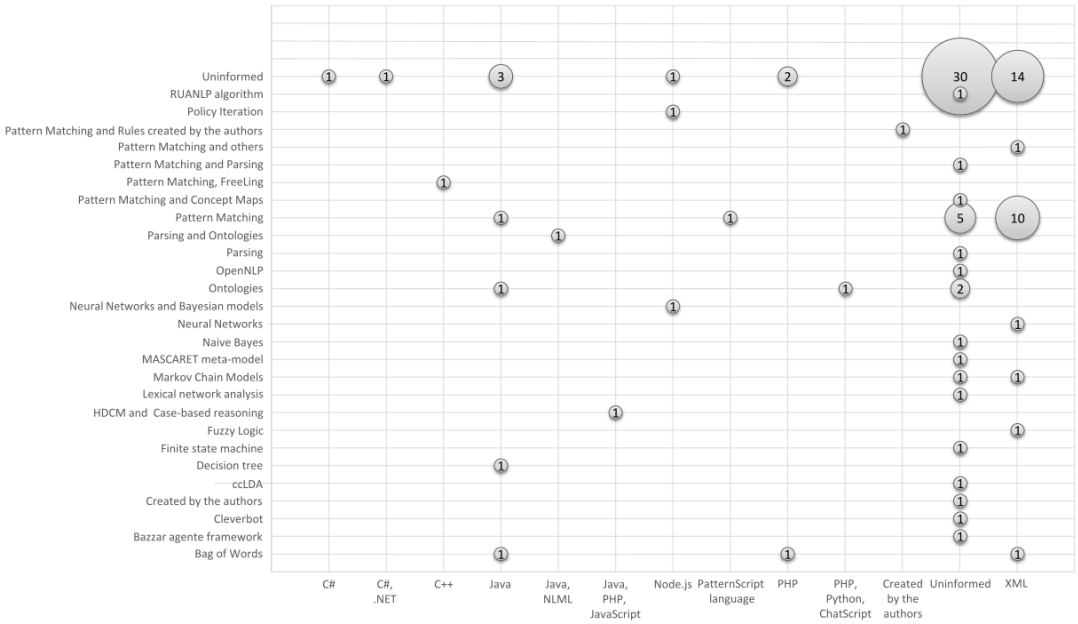


Fig. 6. Association between system logics and programming languages

Multimedia resources. In 97 studies is mentioned that the presented conversational agents talk to the student through text, and some of these use other resources to complement the text (e.g., voice (Sawaki et al. 2008), image (Aguiar et al. 2014)). Two studies did not mention how the conversational agent interacts with the user. In one study, the authors mention that the conversational agent is able to talk through voice, in which it generates the voice through voice synthesizers. In another study, recorded audios are used by the conversational agent during interaction. Figure 7 presents the media that are used by educational conversational agents to interact in with students.

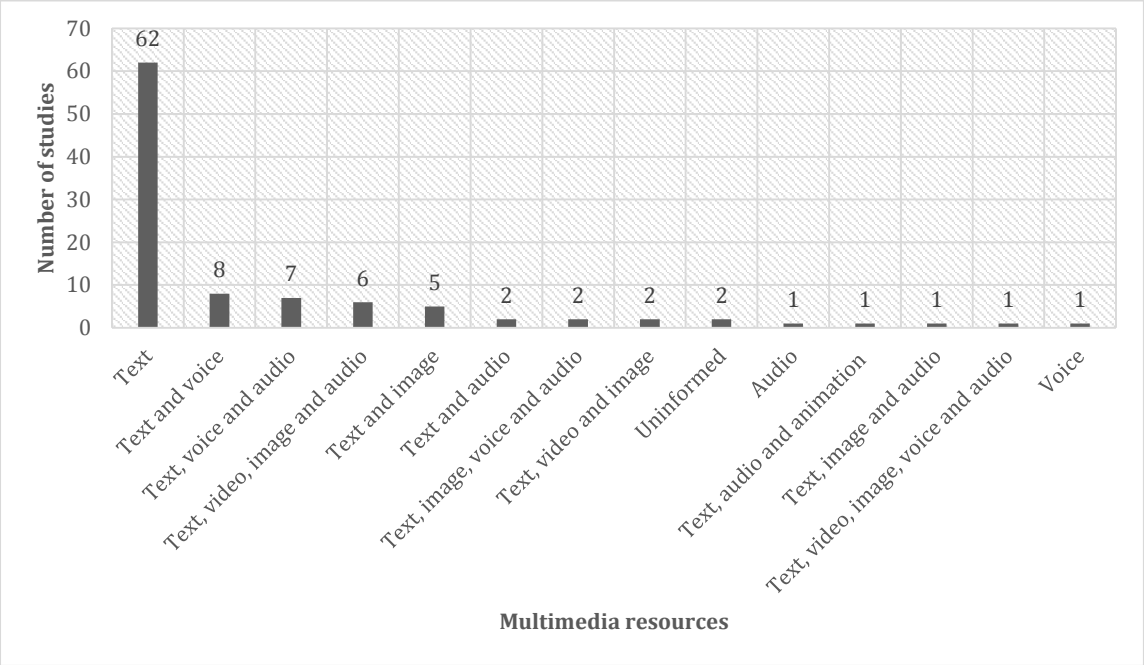


Fig. 7. Conversational agents media

Software process. We have analyzed which software process is commonly used in conversational agents development. We identified that only one study mentioned the use of a process model, and the other 100 studies do not mention it. The study mentioning it just revealed the use of an interactive model of development (Kerly et al. 2008). This result is critical, since it reveals that there is a gap in software development of educational conversational agent. It is known that conversational agents have specific characteristics and properties, so the use of a process model is important to systematize the development.

Educational principle or learning theory. we tried to verify if the educational conversational agents are developed according to some educational principle or learning theory. We observed that 66 primary studies did not mention the use of some educational principle (Table 7). The other studies describe the use of some trends to support their approach, such as Collaborative Learning Theory (11 studies), Bloom's Taxonomy of Educational Objectives (5 studies), Cognitive Load Theory (3 studies), among others.

Table 7
Learning theories and educational principles used in the studies

Theory and educational principles	Number of studies
Uninformed	66
Collaborative learning	11
Educational objectives	5
Cognitive load theory	3
Constructivism	2
Experiential learning theory	2
Social agency theory and cognitive load theory	2
Attribution theory and cognitive evaluation theory	1
Big five theory	1
Conversation theory	1

Deep Learning	1
Knowledge space theory	1
Learning styles	1
Self-regulated learning theory	1
Self-regulated theory e Kolb's learning cycle	1
Zone of proximal development	1
Zone of proximal development and sociocultural theory of learning	1

Graphic representation. This analysis verified how the conversational agents are graphically represented, if so, in which we observed that there is no common sense in the way the conversational agents are visually represented. Table 8 presents the complete data on the conversational agents representation/appearance. We have observed that 29 primary studies describe that the conversational agents do not have appearance, 17 studies do not mention whether the conversational agent has any avatar or image to represent it, and 16 studies describe the appearance of the conversational agent like a full humanoid 3D body. In a smaller proportion, 2D heads (8 studies), humanoid 2D heads (8 studies), fictitious character full 2D body (7 Studies), humanoid 3D head (7 studies), among others, were the choice of representation.

Table 8
Conversational agents appearance

Appearance	**Number of studies**
No image representation	29
Uninformed	17
Full humanoid body 3D	16
Character head 2D	8
Humanoid head 2D	8
Full character body 2D	7
Humanoid head 3D	7
Full humanoid body 2D	4
Character head 3D	1
Character head 3D and character head 2D	1
Full character body 3D	1
Full character body 3D and humanoid head 3D	1
Humanoid head 3D, humanoid head 2D, character head 3D and character head 2D	1

5.4. How the conversational agents used in educational area have been evaluated?

The last research question aims to obtain information about how educational conversational agents are being evaluated. The following points were collected during this analysis: evaluated aspects, empirical methods used, data collection instruments and methods used in the analysis of data.

Evaluation of conversational agents. we noted that a large number of studies (70 studies) uses learning as an evaluation metric. In general, in these studies the authors analyze how much a student learns from the support of a conversational agent. Another aspect evaluated is the affective one. We classified in this category the studies that mention analyzing how motivated and/or satisfied the students felt when interacting with the conversational agent. It is understood

that the focus on analyzing aspects related to the human-computer interaction is more common than the evaluation of the system functionalities. But in some studies (10) the usability is also evaluated. Few initiatives (8 studies) focus on evaluating the quality of the dialogue issue.

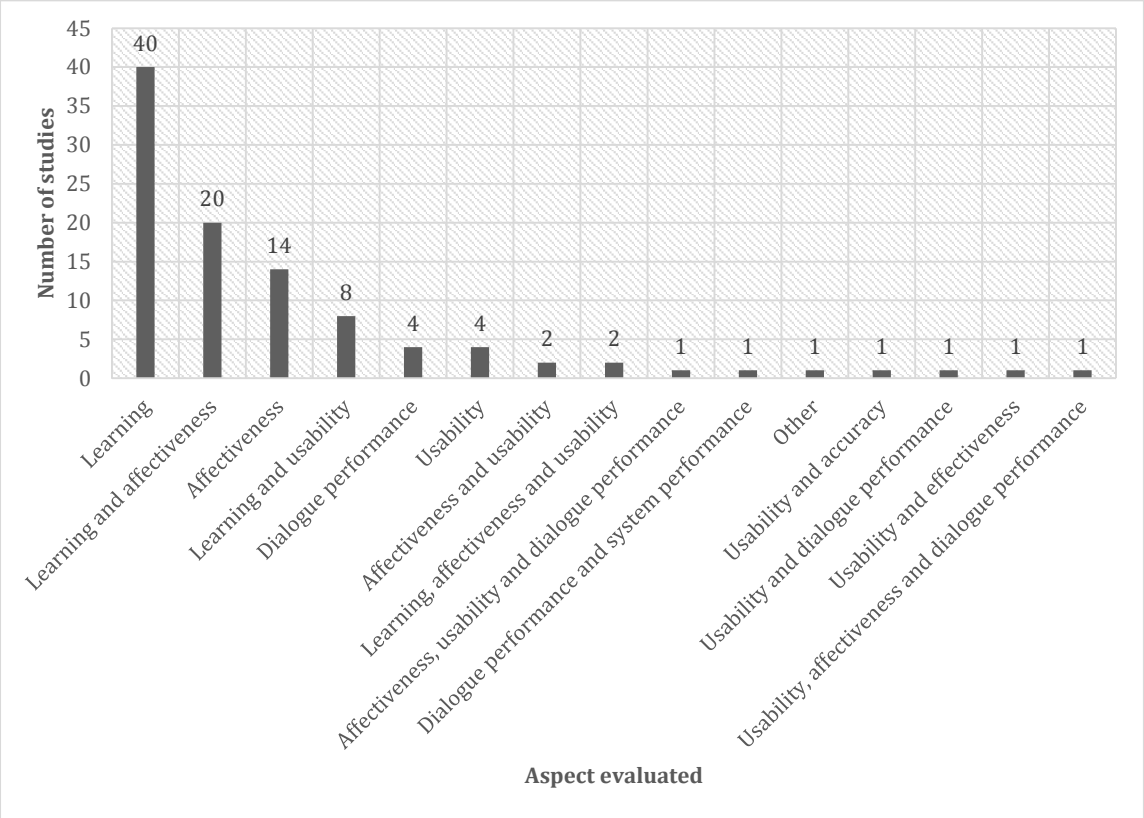


Fig. 8. Main aspects considered during the evaluation of conversational agents

Type of empirical evaluation. We have identified two types of empirical evaluation: case study (30 studies) and experimental study (43 studies). One of them, use case study and experiment study together, as can be seen in Figure 9. We also noted that 27 studies do not mention the method used in the evaluation. This information helps to analyze the maturity of this research area in terms of empirical evaluation.

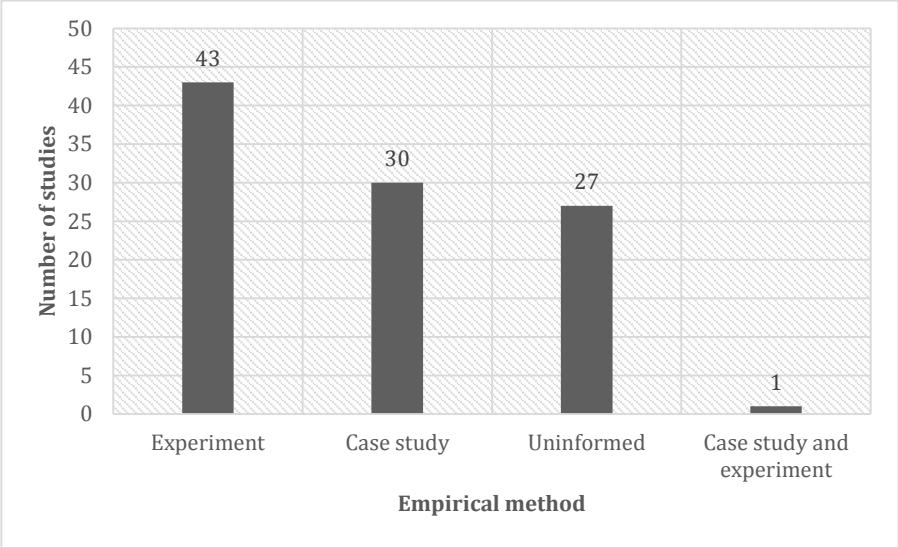


Fig. 9. Type of empirical evaluation of conversational agents

Instruments of the experiments. We also investigated how the studies collect data for empirical analysis. We find that the most studies use questionnaires (41), or questionnaires in conjunction with the analysis of conversation technique (23). In Figure 10 are presented the instruments used for data collection identified in the primary studies.

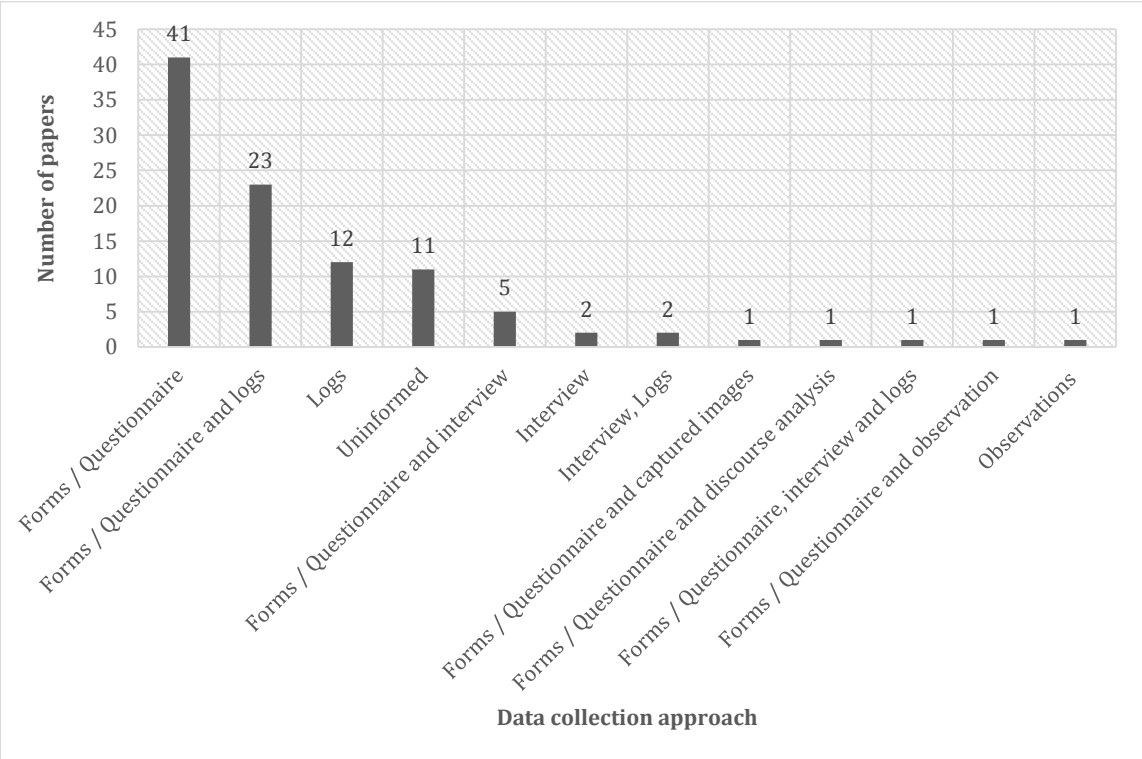


Fig. 10. Instruments used to collect data in the experimental studies

Statistical analysis. we seek to recognize what type of statistical analysis is employed in the experimental studies. More than 50% of the works use inferential statistics (54 studies). A

smaller number uses only descriptive statistics (26). Six studies use these two types of statistical analysis in a combined way. Fifteen works, although they have conducted an evaluation, do not mention the use of statistical analysis.

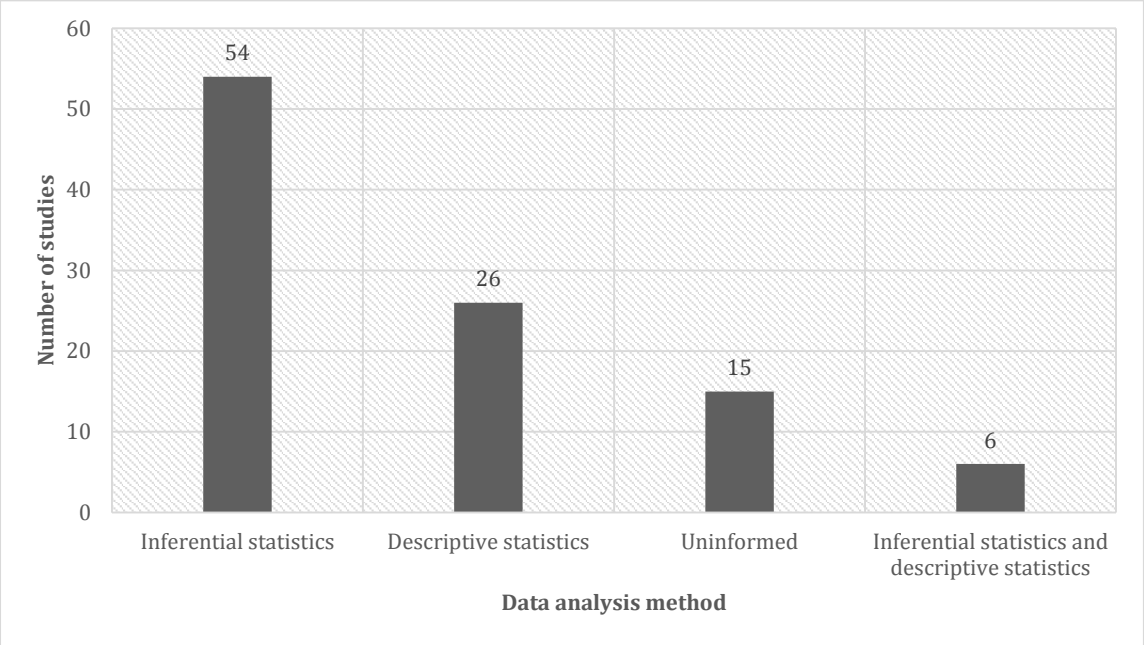


Fig. 11. Type of statistical analysis

6. Discussion

In this section, we present a discussion based on our interpretations and analyses of the results, presenting the main trends and the main gaps that have been observed.

6.1. Areas of Application

In relation to the educational areas and levels of education, we can observe that there is a community that has been concentrating efforts in development of solutions to support higher education, especially for undergraduate students. This result is related to the areas of knowledge in which the conversational agents are being applied, since areas such as engineering and psychology contemplate content that are not seen during elementary school and high school. On the other hand, it is observed that in areas that are taught independently of the educational level, conversational agents have been used (e.g. Physics, Chemistry, Linguistics, Mathematics, Biology, and Computer Science). We also noticed that there are some areas that do not have research on the use of conversational agents (e.g. Administration, Agronomy, Law, among others). Although these areas do not have investigations about this type of support mechanism, we believe that the conversational agents can contribute with them. In the area of law, for example, a conversational agent could represent the role of a witness, a judge, or a prosecutor, to help the student prepare for a trial.

6.2. Development Process

In relation to the development of conversational agents, we can observe that there are several methods/techniques/algorithms that have been used during the implementation, but Pattern Matching constructed with XML is the most used method. We observe that there are a limited number of published studies that explore the new techniques and algorithms of artificial intelligence. We believe that this is a limitation in the area, since most developers use a strategy that was built more than 20 years ago. We also know that one of the main limitations of Pattern Matching is the semantic treatment of language. Due to the advent of methods and techniques for semantic treatment, such as ontologies, we believe that there is a gap in the community regarding this limitation.

Conversational agents are computer programs that seek to imitate human beings, based on natural language interpretation. Due to, they are characterized as good examples of weak AI (Russell and Norving, 2013), because they act as if they were intelligent (intelligence simulation), instead of actually being smart. Based on this, we believe that the development of conversational agents by means of Pattern Matching reflects on the purpose of the researchers to only represent the simulation of intelligence during the interaction, not worrying about whether the conversational agent will learn something with the student.

One of the findings is related to the fact that many studies do not mention the logic of the system and the language used during the implementation of the conversational agents. This information is very important, since it enables the developers and researchers around the world to have a clearer view of the actual functioning of the conversational agent.

Regarding the communication, we noticed that few studies report on the efforts in developing conversational agents that are able to recognize the voice of the humans and interact with them by means of voice. Thus, educational conversational agents continue to be mostly text-based. In this sense, we believe that experimental studies with the intention of finding out whether the communication can influence in the use of conversational agent is an excellent research opportunity.

In relation to development process, we observed an existing gap, since only one study mentioned about the use of a process model. The lack of use of a process model influence on development, especially in the artifacts produced (requirements document, software project, project management plan, test plan and test documents and software architecture), in the software reuse, in the time spent in development, among other aspects. It is possible that current software development models do not provide the necessary support for building chatterbot-type software systems, since they have specific characteristics. We suggest that it is important and necessary to investigate the software development models for conversational agent.

Through our research, we have identified that the authors of the studies, for the most part, are not considering educational principles and tendencies in the development of conversational agents. We believe this happens because they are focusing mainly on increasing the likelihood of use of the conversational agents by students. If no educational theory is adopted, the

application of the conversational agent may be compromised, highly dependent on the approach that is used and on the conduction of the teacher.

Finally, when analyzing the appearance of the conversational agents (considering the final product), we found that most studies do not present and describe about visual aspects of their agents. There are few efforts on the use of some kind of graphical illustration to represent it. Because of the studies that mention about the importance of representation of the conversational agents (Koda et al. 2016, Mehrotra et al. 2016, Ochs et al. 2017), we believe that researchers and developers need to invest more in the appearance of the conversational agents.

6.3. Evaluations

On evaluations issue, we observed that studies focus mainly on analyzing how the interaction with conversational agent influence on student's performance. This type of evaluation, in a certain way, influence on the instruments used, on the type of evaluation performed and on the type of statistical analysis chosen. In this sense, we suggest that the main limitation refers to the evaluation of the conversational agent itself. The studies focus on evaluating the application, but there are few studies that analyze, for example, the quality of chatterbot's dialogue, which refers to the ability of the conversational agent to correctly interpret the message that was transmitted by the student and to present the student with an adequate response. We believe that before the conversational agent is applied to a class, it would be important to carry out the quality of the dialogue evaluation.

The quality question also reflects again on what we observed when analyzing how the conversational agents are developed. The authors do not mention about using a development model, and this fact might influence the quality. We believe that it is important to take quality models and to find out if there are models that enable the quality assessment of conversational agents.

7. Threats to Validity

As a potential threat to validity is related to the studies selected and extracted in the SLR. It is possible that our search string, the chosen databases, or the process adopted for the selection of primary studies may have unintentionally omitted some relevant primary studies. Notwithstanding, we adopted a process for SLR widely recognized in the literature; we chose databases that are recommended by the literature, and the protocol was conferred by four researchers from the area. Given this, we believe that we did not omit important primary studies.

A last potential threat to the validity refers to the findings on the data that have been extracted from each study. To mitigate this threat to validity, to make the analysis more objective possible, we built and used a data extraction online form (available at: < http://bit.ly/2E6muuL>).

8. Conclusions

This study presented an overview on the development and application of conversational agents in formal education, considering from the last ten years, aiming to understand how it is possible to take advantage of this supporting mechanism, and to diffuse its educational use.

The SRL was conducted by four researchers with experience in the area. Through a search in six international databases, 2.619 studies were first identified, and 101 were selected after the application of the inclusion and exclusion criteria. The detailed analysis of these 101 studies brought important results, with the identification of the existents limitations and gaps.

The results of this review may help researchers and teachers in different knowledge areas with an interest in using conversational agents as an educational resource to enable more engaging activities for the students, as well as a guide for the application in educational practice.

As a complementary research, we plan to use the results of this mapping to set guidelines for supporting researchers to develop conversational agents for education. We also plan to investigate software reuse in the context of conversational agents, investigating ways of enhancing software reuse and software development process.

References

Abdul-Kader, S. A., Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6 (7), 72–80.

AbuShawar, B., Atwell, E. (2016) Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora. *Workshops and Session Series on Chatbots and Conversational Agents*, 29-38.

Aguiar, E. V. B., Tarouco, L. M. R., Reategui, E. (2014). Supporting problem-solving in Mathematics with a conversational agent capable of representing gifted students' knowledge. *47th Hawaii International Conference on System Science*, 130-137.

Basili, V. R. (1992). Software Modeling and Measurement: The Goal/Question/Metric Paradigm. https://drum.lib.umd.edu/handle/1903/7538. College Park, MD, USA.

Bloom, B. S. (1984). The 2 Sigma problem: The Research for Methods of Group Instruction as Effective as One-to-one Tutoring. *Educational Researcher*, 13 (6), 4-16.

Business Insider. (2016). Facebook has opened up analytics and developer tools for Messenger bots, BI Intelligence. http://www.businessinsider.com/facebook-opens-analytics-developer-tools-for-messenger-bots-2016-11.

Chatbot's Magazine. (2017). 6 Ways Artificial Intelligence and Chatbots Are Changing Education. https://chatbotsmagazine.com/six-ways-a-i-and-chatbots-are-changing-educationc22e2d319bbf.

Cleverscript. (2018). Data for Machine Learning. Conversation Data API. https://www.existor.com/products/cleverbot-data-for-machine-learning/.

Crown, S., Fuentes, A., Jones, R., Nambiar, R., Crown, D. (2010). Anne G. Neering: Interactive Chatbot to Motivate and Engage Engineering Students. *ASEE Annual Conference and Exposition*, 15.181.1 - 15.181.13.

de Gasperis, G., Chiari, I., Florio, N. (2013). AIML Knowledge Base Construction from Text Corpora. In: Yang XS. (eds) *Artificial Intelligence, Evolutionary Computing and Metaheuristics. Studies in Computational Intelligence*, 427 (1), 287-318, Springer, Berlin, Heidelberg.

Dyba, T., Dingsoyr T., Hanssen, G. K. (2007). Applying Systematic Reviews to Diverse Study Types: An Experience Report. *International Symposium on Empirical Software Engineering and Measurement*, 225-234.

Fletcher, J. D. (2001). Technology, the Columbus Effect, and the Third Revolution in Learning, https://apps.dtic.mil/dtic/tr/fulltext/u2/a406772.pdf

Følstad, A., Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *ACM Interactions*, 24(4), 38-42.

Fryer, L. K., Ainley, M., Thompson, A., Gibson, A., Sherlock, Z. (2017). Stimulating and sustaining interest in a language course: An experimental comparison of Chatbot and Human task partners. *Computers in Human Behavior*, 75(1), 461-468.

Ghose, S., Barua, J. J. (2013). Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor. *International Conference on Informatics, Electronics and Vision*, 1-5.

Graesser, A. C., Li, H., Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*, 23(5), 374-380.

Juniper Research. (2017). Chatbots Infographic - Key Statistics. https://www.juniperresearch.com/resources/infographics/chatbots-infographic-key-statistics-2017.

Jurafsky, D., Martin, J. H. (2018). Dialog Systems and Chatbots. *Speech and Language Processing* (3rd ed. draft), 1-26.

Kane, D. (2016). The Role of Chatbots in Teaching and Learning. In: Scott, R., Gregor, M. N. (Eds.), *E-Learning and the Academic Library: Essays on Innovative Initiatives.* McFarland, 131-147.

Kasap, Z., Magnenat-Thalmann, N. (2008). Intelligent Virtual Humans with Autonomy and Personality: State-of-the-Art. Magnenat-Thalmann. N. et al. (Eds.): *New Advances in Virtual Humans*, 43–84.

Kitchenham, B.A., Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering. Keele University and Durham University, UK,2007 Technical report. EBSE 2007-001.

Koda, T., Ishioh, T., Watanabe, T., Kubo, Y..(2016). Perception of Masculinity and Femininity of Agent's Appearance and Self-adaptors. *8th International Conference on Agents and Artificial Intelligence*, 3-18.

Lee, C., Jung, S., Kim, S., Lee, G. G. (2009). Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51 (5), 466–484.

Lin, L., Atkinson, R. K., Christopherson, R. M., Joseph, S. S., Harrison, C. J. (2013). Animated agents and learning: Does the type of verbal feedback they provide matter?. *Computers & Education*, 67 (1), 239-249.

Masche, J., Le, N. T. (2017). A Review of Technologies for Conversational Systems. *5th International Conference on Computer Science, Applied Mathematics and Applications*, 212-225.

Mateas, M.; Stern, A. (2004). Natural language understanding in Façade: Surface-text processing. *2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 3-13.

Mayer, R. E., Dow, G.T., Mayer, S. (2003). Multimedia learning in an interactive self-explaining environment: What works in the design of agent-based microworlds? *Journal of Educational Psychology*, 95 (4), 806-813.

Mdehn, D., Mulken, S. (2000). The impact of animated interface agents: a review of empirical research. *International Journal of Human-Computer Studies*, 52 (1), 1-22.

Mehrotra, S., Motti, V.G., Frijns, H., Akkoc, T., Yengeç, S.B., Calik, O., Peeters, M.M.M., Neerincx, M.A.. (2016). Embodied conversational interfaces for the elderly user. *8th Indian Conference on Human Computer Interaction*, 90-95.

Moreno, R., Mayer, R. E., Spires, H. A., Lester, J. C. 2001. The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents? *Cognition and Instruction*, 19 (2), 177-213.

Murtiningsih, S. R. (2011). Issues in Teaching English to Speakers of Other Languages (Tesol): A Review Of Literature. *Jurnal Lingua Cultura*, 5(1), 53-60.

Ochs, M., Pelachaud, C., McKeown, G. (2017). A user perception-based approach to create smiling embodied conversational agents. *ACM Transactions on Interactive Intelligent Systems*, 7 (1), 4:1 - 4:33.

Pereira Júnior, C., Francisco, R., Silva, L., Veiga, E., Fernandes, M., Dorça, F. (2017). Use of Ontologies for Conversational Agents in the Context of Teaching-Learning: A Systematic Review of Literature (in Portuguese). *27th Brazilian Symposium on Computers in Education*, 183-192.

Perez-Marin, D.; Pascual-Nieto, I. (2011). *Conversational Agent and Natural Language Interaction: Techniques and Effectives Practices*. USA: IGI Global.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M. (2008). Systematic mapping studies in software engineering. *12th International conference on Evaluation and Assessment in Software Engineering*, 68-77.

Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1-18.

Radziwill, N. M., Benton, M. C. (2017). Evaluating Quality of Chatbots and Intelligent Conversational Agents. *Software Quality Professional; Milwaukee*, 19(3), 25-36.

Rickel, J. (2001). Intelligent Virtual Agents for Education and Training: Opportunities and Challenges. *3rd International Workshop on Intelligent Virtual Agents*, 15-22.

Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited, 2013.

Sawaki, M., Minami, Y., Higashinaka, R., Dohsaka, K. and Maeda, E.. (2008). “Who is this” quiz dialogue system and users' evaluation. IEEE Spoken Language Technology Workshop, page(s): 149-152.

Schmidt, R. W. (2018). Learning System Customer Service Chatbot. School of Computer Science Technical Reports, 1-5. https://smartech.gatech.edu/handle/1853/59561

Schroeder, N. L., & Adesope, O. O. (2014). A Systematic Review of Pedagogical Agents’ Persona, Motivation, and Cognitive Load Implications for Learners. *Journal of Research on Technology in Education*, 46 (3), 229-251.

Schroeder, N. L., Adesope, O. O., Gilbert, R. B. (2013). How Effective are Pedagogical Agents for Learning? A Meta-Analytic Review. *Journal of Educational Computing Research*, 49 (1), 1-39.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., Pineau, J. (2016). Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *30th AAAI Conference on Artificial Intelligence*, 3776–3784.

Shang, L., Lu, Z., Li, H. (2015). Neural responding machine for short-text conversation. *7th International Joint Conference on Natural Language Processing*, 1577-1586.

Wallace, R. (1995). ALICE: Artificial Linguistic Internet Computer Entity: The A.L.I.C.E A.I. Foundation. http://alicebot.blogspot.com.br/.

Weizenbaum, J. (1996). ELIZA A computer program for the study of natural language communication between man and machine. *Communications of the ACM,* 10(8), 36–45.

Wen, T., Vamdyke, D., Mrkšic´, N., Gašic´, M., Rojas-Barahona, L. M., Su, P., Ultes, S., Young, S. A network-based end-to-end trainable taskoriented dialogue system. *15th Conference of the European Chapter of the Association for Computational Linguistics,* 438-449.

Wilcox, B. (2011). Pattern Matching for Natural Language Applications. https://github.com/bwilcox-1234/ChatScript/blob/master/WIKI/PAPERS/Paper-Pattern-Matching-for-Natural-Language-Applications.md.

Winkler, R., Söllner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. *78th Annual Meeting of the Academy of Management*, 1-40.

Zumstein, D.; Hundertmark, S. (2017). Chatbots--an interactive technology for personalized communication, transactions and services. *IADIS International Journal on WWW/Internet*, 15(1), 96-109.

ICMC USP
SÃO CARLOS