UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Uma linha de produtos de software orientada a serviços para
sistemas de auxílio à visitação de museus**

Ricardo Fuzeto

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

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Ricardo Fuzeto

Uma linha de produtos de software orientada a serviços
para sistemas de auxílio à visitação de museus

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. Rosana Teresinha Vaccare Braga

USP – São Carlos
Julho 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)

F9491 Fuzeto, Ricardo
Uma linha de produtos de software orientada
a serviços para sistemas de auxílio à visitaço de
museus / Ricardo Fuzeto; orientadora Rosana
Teresinha Vaccare Braga. -- So Carlos, 2019.
128 p.

Dissertao (Mestrado - Programa de Ps-Graduao
em Cincias de Computao e Matemtica
Computacional) -- Instituto de Cincias Matemticas
e de Computao, Universidade de So Paulo, 2019.

1. Linhas de produtos de software. 2. Servios.
3. Informtica na educao. 4. Museus. 5. Internet
das coisas. I. Teresinha Vaccare Braga, Rosana,
orient. II. Ttulo.

Ricardo Fuzeto

**A service-oriented software product line for museum
visitation assistance systems**

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. Rosana Teresinha Vaccare Braga

USP – São Carlos
July 2019

Aos meus pais, irmão e amigos.

AGRADECIMENTOS

Primeiro, agradeço aos meus pais pelo apoio dado. Sem o suporte deles, eu não teria tido a chance de realizar este trabalho.

À minha orientadora, Rosana Teresinha Vaccare Braga, por me aceitar como aluno e me ajudar a aprender mais sobre sua área de domínio. Também a agradeço pela perseverança em acreditar no meu potencial, mesmo nos momentos em que senti ter lhe decepcionado. Carregarei as lembranças desta fé em minha memória, pelo restante da minha vida.

A Venilton Falvo Junior por trabalhar em conjunto conosco, apoiando as atividades realizadas, fornecendo suporte em relação ao seu trabalho anterior publicado e utilizado como ponto de partida para este trabalho.

Aos membros participantes da banca avaliadora por doarem parte do seu tempo de trabalho para participarem da avaliação deste trabalho, e por fornecerem uma fração de seus conhecimentos para melhorar ainda mais o trabalho aqui apresentado.

Aos docentes do ICMC, em especial aqueles que trabalham junto ao Laboratório de Engenharia de Software (LabES), pelas discussões realizadas e pelas opiniões dadas visando o enriquecimento deste trabalho.

À USP e ao ICMC pela vaga disponibilizada no Programa de Pós-Graduação em Ciências da Computação e Matemática Computacional, principalmente por ser uma vaga de bolsista. Também os agradeço pela infra-estrutura oferecida, e pelo acesso facilitado aos docentes do instituto.

Ao Procad-Capes pela oportunidade de me conectar com os pesquisadores envolvidos no projeto. Também agradeço pela oportunidade que me foi dada de visitar o Núcleo de Excelência em Tecnologias Sociais (NEES) da UFAL, em Maceió/AL, a convite do Prof. Rafael de Amorim Silva.

Ao CNPq pela oportunidade dada através da bolsa de estudos disponibilizada, que possibilitou a condução das pesquisas aqui apresentadas.

*“A vida não se trata do quanto você consegue bater.
A vida se trata do quanto você consegue suportar apanhar,
e continuar seguindo em frente. É disso que a vitória é feita.”
(Sylvester Stallone em Rocky 6)*

RESUMO

FUZETO, R. **Uma linha de produtos de software orientada a serviços para sistemas de auxílio à visitaço de museus**. 2019. 128 p. Dissertao (Mestrado em Cincias – Cincias de Computao e Matemtica Computacional) – Instituto de Cincias Matemticas e de Computao, Universidade de So Paulo, So Carlos – SP, 2019.

Os sistemas de auxlio visitao de museus apresentam um grau de reso de *software* baixo, dado que o volume de trabalhos publicados contendo a proposta de um novo sistema e/ou modelagem consideravelmente elevado, e com grau de refereniao entre tais trabalhos prximo de zero. Alm disso, diversos estudos neste domnio de aplicaes utilizam tcnicas de cincia de contexto e fazem uso de dispositivos de Internet das Coisas (IoT), o que agrava o grau de reso devido variedade de sensores e dispositivos existentes. Este trabalho apresenta a *Museum Touring and Tutoring Product Line* (MTTPL), linha de produtos orientada a servios para o domnio dos sistemas de auxlio visitao de museus. Os requisitos desta linha de produtos foram coletados principalmente dos resultados de um mapeamento sistemtico feito sobre sistemas *E-learning* e dispositivos IoT. Como forma de avaliao preliminar da linha de produtos proposta, foi realizada uma prova de conceito em que a MTTPL foi instanciada, dando origem ao SmartICMC, um museu virtual para apoiar a visitao ao Museu da Fauna e Flora do ICMC-USP. A prova de conceito fornece indcios de que a linha de produtos proposta vlida e aplicvel para sistemas de auxlio visitao de museus. Como principal contribuio, espera-se que a MTTPL possa ser utilizada para instanciar outros sistemas deste domnio, aumentando o reso dos *assets* providos pela linha de produtos.

Palavras-chave: Linha de produtos, servios, U-learning, museus, Internet das coisas.

ABSTRACT

FUZETO, R. **A service-oriented software product line for museum visitation assistance systems.** 2019. 128 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.

Museum visitation assisting systems present a low software reuse degree, given that the amount of published papers containing a proposal for a new system and/or modeling is considerably elevated, and with a referring degree among those works near to zero. Besides, many studies on this application domain use context awareness techniques and Internet of Things (IoT) devices, aggravating software reuse degree due to the variety of existing IoT devices. This work presents the Museum Touring and Tutoring Product Line (MTTPL), a service-oriented product line for the museum visitation assisting systems domain. Requirements for this product line were collected mainly from the results of a systematic mapping about E-learning systems and IoT devices. As a preliminary evaluation of the proposed product line, a proof of concept was held in which MTTPL was instantiated, giving rise to SmartICMC, a virtual museum to support the visit to the ICMC-USP Museum of Fauna and Flora. The proof of concept provided evidence that the proposed product line is valid and applicable to museum visitation support systems. As a major contribution, it is expected that MTTPL can be used to instantiate other systems in this domain, increasing the reuse of the assets provided by the product line.

Keywords: Product lines, services, U-learning, museums, internet of things.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura de um serviço, adaptada de Rosen <i>et al.</i> (2008).	35
Figura 2 – As três atividades principais do ciclo de vida de uma LPS, segundo Clements e Northrop (2002).	42
Figura 3 – Exemplo de modelo de <i>features</i>	44
Figura 4 – Engenharia de LPS usando o ESPLEP (GOMAA, 2005) (fonte: (DONEGAN, 2008)).	44
Figura 5 – Ano de publicação dos estudos selecionados.	55
Figura 6 – País de origem dos estudos selecionados.	56
Figura 7 – Ambientes de aprendizagem dos estudos selecionados.	57
Figura 8 – Tipos de contribuição dos estudos selecionados.	58
Figura 9 – Recursos educacionais utilizados nos estudos selecionados.	59
Figura 10 – Tecnologias IoT utilizadas nos estudos selecionados.	60
Figura 11 – Diagrama de casos de uso da MTTPL.	69
Figura 12 – Modelo conceitual da MTTPL.	70
Figura 13 – Modelo de <i>features</i> da MTTPL.	74
Figura 14 – Organização dos serviços da MTTPL.	78
Figura 15 – Classes base do Broker de Sensores.	79
Figura 16 – Classes base do Recomendador de Itens.	79
Figura 17 – Classes base do Provedor de Conteúdo.	80
Figura 18 – Classes base do Editor de Conteúdos.	81
Figura 19 – Classes base do Mapeador de Rotas.	81
Figura 20 – Classes base do Provedor de Usuários.	82
Figura 21 – Classes base do Provedor de Avaliações.	83
Figura 22 – Classes base do Localizador.	83
Figura 23 – Classes base do Backend.	84
Figura 24 – Classes base do Aplicativo Móvel.	85
Figura 25 – Modelo de <i>features</i> do SmartICMC, a partir do modelo da MTTPL.	90
Figura 26 – Serviços do SmartICMC, a partir do modelo definido pela MTTPL.	91
Figura 27 – Aplicativo Móvel do SmartICMC.	91
Figura 28 – Mapeador de Rotas do SmartICMC.	92
Figura 29 – Provedor de Conteúdo do SmartICMC.	92
Figura 30 – Backend do SmartICMC.	92
Figura 31 – Provedor de Usuários do SmartICMC.	93

Figura 32 – Provedor de Usuários do SmartICMC após a modelagem dos requisitos faltantes.	97
Figura 33 – Provedor de Conteúdo do SmartICMC após a modelagem dos requisitos faltantes.	98
Figura 34 – Mapeador de Rotas do SmartICMC após a modelagem dos requisitos faltantes.	98
Figura 35 – Backend do SmartICMC após a modelagem dos requisitos faltantes.	99
Figura 36 – Aplicativo Móvel do SmartICMC após a modelagem dos requisitos faltantes.	100
Figura 37 – <i>Brokers</i> do <i>Backend</i> instanciados pelo <i>RestController</i>	101
Figura 38 – Código-fonte do método <i>getMapNodes</i> da classe <i>RoutePlanner</i>	101
Figura 39 – Código-fonte do método <i>getItem</i> da classe <i>ItemDao</i>	102
Figura 40 – Código-fonte do método <i>logoutUser</i> da classe <i>RestController</i>	103
Figura 41 – Capturas de tela do SmartICMC.	106
Figura 42 – Características do SmartICMC, representadas em código-fonte através da classe <i>Features</i>	107
Figura 43 – Bússola utilizada pelo Aplicativo Móvel para auxiliar a navegação.	109

LISTA DE TABELAS

Tabela 1 – Termos e sinônimos	52
Tabela 2 – Dados coletados dos estudos	52
Tabela 3 – Resultados da primeira consulta às bases de dados	53
Tabela 4 – Estudos selecionados por etapa de leitura.	55

LISTA DE ABREVIATURAS E SIGLAS

CRUD	<i>Create, Read, Update and Delete</i>
CRUD	<i>Create, Read, Update e Delete</i>
DAO	<i>Data Access Object</i>
DI	injeção de dependências, ou <i>Dependency Injection</i>
DTO	<i>Data Transfer Object</i>
E-learning	Aprendizado eletrônico
EDLC	<i>Evolutionary Domain Life Cycle</i>
ESPLEP	<i>Evolutionary Software Product Line Engineering Process</i>
FAST	<i>Family-Oriented Abstraction, Specification and Translation</i>
FODA	<i>Feature-Oriented Domain Analysis</i>
HOC	componente de alta ordem, ou <i>High Order Component</i>
IoT	Internet das Coisas, ou <i>Internet of Things</i>
LabES	Laboratório de Engenharia de Software
LMS	“ <i>Learning Management System</i> ” ou “Sistema de Gerenciamento de Aprendizado”
LMSs	<i>Learning Management Systems</i>
LPOS	Linha de Produtos Orientada a Serviços
LPOS	Linha de Produtos Orientada a Serviços
LPS	Linhas de Produtos de Software
M-learning	Aprendizado móvel
MTTPL	<i>Museum Touring and Tutoring Product Line</i>
MTTPL	<i>Museum Touring and Tutoring Product Line</i>
MVP	Produto Mínimo Viável, ou <i>Minimum Viable Product</i>
NEES	Núcleo de Excelência em Tecnologias Sociais
ORM	Mapeador Objeto Relacional, ou <i>Object Relational Mapper</i>
P-learning	Aprendizado pervasivo
PDA	<i>Portable Digital Assistant</i>
PLUS	<i>Product Line UML-based Software Engineering</i>
QoS	Qualidade de Serviço, ou <i>Quality of Service</i>
QR Codes	Códigos <i>Quick Response</i>
REST	<i>REpresentational State Transfer</i>
RPC	<i>Remote Procedure Call</i>

SaaS	<i>Software as a Service</i>
SLA	Acordo de Nível de Serviço ou <i>Service Level Agreement</i>
SLR	Revisão Sistemática
SM	Mapeamento Sistemático, ou <i>Systematic Mapping</i>
SM	Mapeamento Sistemático, ou <i>Systematic Mapping</i>
SO	Sistema Operacional
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SPA	<i>Single Page Application</i>
U-learning	Aprendizado ubíquo
UML	Unified Modeling Language
URI	<i>Uniform Resource Identifier</i>
UX	Experiência de Usuário, ou <i>User Experience</i>
WADL	<i>Web Application Description Language</i>
WSDL	<i>Web Services Description Language</i>
WSNs	Redes de Sensores Sem-fio

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização e Motivação	25
1.2	Problema em Destaque	27
1.3	Solução Desenvolvida	28
1.4	Estrutura do Documento	28
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	Learning Management Systems	31
2.1.1	<i>Tipos de sistemas de aprendizado</i>	32
2.1.2	<i>Contexto</i>	33
2.2	Internet das Coisas - IoT	33
2.2.1	<i>Exemplos de aplicações de IoT</i>	34
2.3	Serviços de Software	35
2.3.1	<i>Propriedades de serviços</i>	36
2.3.2	<i>Tipos de serviço</i>	37
2.3.3	<i>Web services</i>	38
2.3.3.1	<i>SOAP</i>	39
2.3.3.2	<i>REST</i>	39
2.4	Linhas de Produtos de Software	40
2.4.1	<i>Variabilidade</i>	41
2.4.2	<i>Features</i>	41
2.4.3	<i>Product Line UML-based Software Engineering</i>	43
2.4.4	<i>Engenharia de Aplicação e Engenharia de Domínio</i>	45
2.5	LPS x SOA	45
2.5.1	<i>Diferenças entre LPS, SOA e LPOS</i>	46
2.6	Considerações Finais	47
3	MAPEAMENTO SISTEMÁTICO	49
3.1	Planejamento	50
3.1.1	<i>Objetivos e Questões de Pesquisa</i>	50
3.1.2	<i>String genérica e dados coletados</i>	51
3.1.3	<i>Bases de dados e critérios de inclusão/exclusão</i>	52
3.2	Condução	53

3.2.1	<i>Primeira etapa</i>	53
3.2.2	<i>Segunda etapa</i>	54
3.2.3	<i>Terceira etapa</i>	54
3.3	Sumarização dos Dados	54
3.3.1	<i>Ano</i>	54
3.3.2	<i>País</i>	55
3.3.3	<i>Ambientes de aprendizagem</i>	56
3.3.4	<i>Tipos de contribuição</i>	57
3.3.5	<i>Recursos educacionais</i>	58
3.3.6	<i>Tecnologias IoT</i>	58
3.3.7	<i>Utilização de serviços</i>	59
3.4	Características dos Estudos Seleccionados	60
3.4.1	<i>Middleware</i> s	61
3.4.2	<i>Idiomas</i>	61
3.4.3	<i>Computação e Mecatrônica</i>	62
3.4.4	<i>Smart Campus</i>	62
3.4.5	<i>Visitação de Museus</i>	63
3.5	Considerações Finais	64
4	A LINHA DE PRODUTOS: MTTPL	65
4.1	Definição	66
4.2	Processo Adotado	67
4.3	Requisitos	67
4.4	Diagrama de Casos de Uso	69
4.5	Modelo Conceitual	69
4.6	<i>Features</i>	70
4.7	Serviços	73
4.7.1	<i>Definição dos Serviços</i>	73
4.7.2	<i>Arquitetura Geral</i>	76
4.8	Classes-Base	78
4.8.1	<i>Entidades Comuns</i>	78
4.8.2	<i>Broker de Sensores</i>	78
4.8.3	<i>Recomendador de Itens</i>	79
4.8.4	<i>Provedor de Conteúdo</i>	80
4.8.5	<i>Editor de Conteúdos</i>	80
4.8.6	<i>Mapeador de Rotas</i>	81
4.8.7	<i>Provedor de Usuários</i>	82
4.8.8	<i>Provedor de Avaliações</i>	82
4.8.9	<i>Localizador</i>	82
4.8.10	<i>Backend</i>	83

4.8.11	<i>Aplicativo Móvel</i>	84
4.9	<i>Considerações Finais</i>	84
5	SMARTICMC	87
5.1	<i>Descrição Geral</i>	87
5.2	<i>Objetivo</i>	88
5.3	<i>Engenharia do Sistema</i>	89
5.3.1	<i>Features</i>	89
5.3.2	<i>Serviços</i>	89
5.3.3	<i>Classes do SmartICMC</i>	90
5.3.4	<i>Tecnologias de Implementação</i>	93
5.3.4.1	<i>Java</i>	93
5.3.4.2	<i>Spring Boot</i>	94
5.3.4.3	<i>MySQL</i>	94
5.3.4.4	<i>jOOQ</i>	94
5.3.4.5	<i>JavaScript</i>	95
5.3.4.6	<i>React</i>	95
5.3.4.7	<i>Webpack e Babel</i>	96
5.3.4.8	<i>Plataformas de Implantação</i>	96
5.4	<i>Organização do Sistema</i>	96
5.4.1	<i>Provedor de Usuários</i>	97
5.4.2	<i>Provedor de Conteúdo</i>	97
5.4.3	<i>Mapeador de Rotas</i>	97
5.4.4	<i>Backend</i>	98
5.4.5	<i>Aplicativo Móvel</i>	99
5.5	<i>Código-fonte Derivado</i>	100
5.5.1	<i>Backend - RestController</i>	100
5.5.2	<i>Mapeador de Rotas - RoutePlanner</i>	101
5.5.3	<i>Provedor de Conteúdo - ItemDao</i>	101
5.5.4	<i>Provedor de Usuários - RestController</i>	102
5.6	<i>Considerações Finais</i>	103
6	RESULTADOS	105
6.1	<i>Assets Implementados</i>	105
6.2	<i>Classes</i>	106
6.3	<i>Objetivos do SmartICMC</i>	108
6.4	<i>Objetivos da MTTP</i>	108
6.5	<i>Conclusões</i>	109
7	CONCLUSÃO	111

7.1	Contribuições	111
7.2	Dificuldades e Limitações	112
7.3	Trabalhos Futuros	113
REFERÊNCIAS		115
GLOSSÁRIO		123
APÊNDICE A	ESTUDOS SELECIONADOS PELO SM	125
APÊNDICE B	ESTUDOS MAIS RELEVANTES DO SM	127

INTRODUÇÃO

A educação evoluiu, em tempos recentes, com a crescente preocupação em manter os estudantes interessados nas diversas atividades de aprendizado. Tal evolução trouxe, também, adaptações nos processos de ensino, o que exigiu a adequação dos professores aos novos métodos.

Uma das causas da evolução da educação é a presença de novas tecnologias, que, ao desempenharem o papel de possibilitadoras, estendem a gama de possibilidades de se aplicar tecnologias nas atividades de aprendizado e ensino. Pode-se citar como exemplo deste impacto a popularização dos projetores multimídia, permitindo que os docentes se utilizassem de conteúdos eletrônicos diversos para a condução de aulas, sem a dependência total do quadro-negro. A aplicação de projetores multimídia nas salas de aula fez com que docentes pudessem preparar aulas previamente, utilizando conjuntos de *slides*, chegando, em alguns casos, a dispensar totalmente o quadro-negro.

O paradigma da Internet das Coisas, ou *Internet of Things* (IoT), com o grau de popularização ganho na última década, promete impactar significativamente o domínio dos sistemas educacionais. Os dispositivos computacionais utilizados em IoT são caracterizados pela inteligência, ciência do contexto e independência. Assim, a IoT trata da disposição de sensores pelos ambientes de um local onde se deseja medir certos aspectos, a fim de tornar um ou mais serviços mais eficientes. Este paradigma promete ser uma evolução da computação contemporânea, fazendo com que sensores e sistemas de informação possam ser utilizados para automatizar atividades de rotina, aumentar a eficiência de alguns serviços, e, também, melhorar a qualidade de vida das pessoas envolvidas ([ATZORI; IERA; MORABITO, 2010](#)).

1.1 Contextualização e Motivação

O uso de novas tecnologias para sistemas educacionais tem sido estudado com interesse crescente. É de se esperar que a IoT também esteja sendo adaptada para a educação. Alguns tra-

balhos nesta área se utilizam de tecnologias já associadas à IoT, como etiquetas RFID (CHANG; YEY; CHENG, 2010; CHU; HWANG, 2010; CHENG *et al.*, 2005) e Redes de Sensores Sem-fio (WSNs) (SILVA *et al.*, 2009; HE *et al.*, 2008; KAISER; OERTEL, 2006). No entanto, alguns estudos focam, também, no uso dos *smartphones* como tecnologia IoT fundamental (GONZÁLEZ *et al.*, 2015; OGATA, 2013).

O uso dos *smartphones* começou com o surgimento do paradigma do aprendizado móvel (M-learning). Este paradigma traz mobilidade aos processos de ensino e aprendizado, permitindo que alunos e professores possam exercer atividades pedagógicas sem a dependência de um local projetado especificamente para tal, assim podendo construir os próprios ambientes de aprendizado (GONZÁLEZ *et al.*, 2015; GONZÁLEZ *et al.*, 2014; ZAPPATORE *et al.*, 2016). Com isso, o M-learning trouxe uma visão inicial para alunos e docentes da possibilidade de aprender e ensinar de forma personalizada, com objetos de aprendizado realísticos.

Os sistemas de aprendizado ubíquo (U-learning) aparecem com a utilização de tecnologias que sejam efetivamente IoT. Não se trata apenas de sensores capazes de coletar dados sobre o ambiente e os usuários, o sistema é capaz de utilizar tais informações para tornar as experiências de aprendizado totalmente personalizadas, e também adequadas à situação dos usuários. Assim, começa-se a criar uma preocupação com o *contexto* dos usuários (XUE; WANG; CHEN, 2011).

Dentre os diferentes domínios de aplicação de sistemas de aprendizagem, pode-se destacar os de auxílio à visitação de museus. Visitantes de um museu podem aprender mais sobre os itens em exibição, quando o museu é projetado para apresentar-lhes informações mais detalhadas sobre tais itens. Assim, é possível que um museu seja planejado para que atividades de aprendizagem sejam conduzidas dentro do mesmo. Portanto, é possível implantar um sistema de aprendizado eletrônico visando aprimorar tais atividades.

Existem diversos estudos aplicando os conceitos de IoT no domínio da educação, visando o desenvolvimento de ambientes de aprendizado M-learning e U-learning. Esses estudos foram identificados através de um mapeamento sistemático, apresentado no Capítulo 3. Dentre os estudos selecionados, há um grande volume de características semelhantes que são retrabalhadas para cada sistema apresentado, aumentando, assim, o trabalho desnecessário realizado a cada estudo. Isto indica a falta de estudos neste tema visando a aplicação de técnicas de reúso de software, como Linhas de Produtos de Software (LPS) (GOMAA, 2005; CLEMENTS; NORTHROP, 2002) e serviços (ROSEN *et al.*, 2008; VERBORGH *et al.*, 2011).

As LPS auxiliam o reúso de soluções em software através da definição de um conjunto de sistemas com características comuns e gerenciáveis, pertencentes a um mesmo segmento de mercado e que sejam desenvolvidos a partir de um mesmo conjunto de recursos-chave (CLEMENTS; NORTHROP, 2002). Assim, uma LPS identifica as características comuns, opcionais e alternativas em um domínio de aplicação, possibilitando a geração de novos produtos com configurações diferentes a partir da escolha das características desejadas.

Um bom exemplo da utilização de LPS em sistemas de aprendizagem é o trabalho de (JUNIOR, 2015), que apresenta o desenvolvimento e avaliação de uma linha de produtos voltada a sistemas de aprendizagem móveis, denominada *M-SPLearning*. Esta LPS foi elaborada com base em 18 estudos selecionados após uma revisão sistemática, em que se apresenta uma LPS para o domínio de *M-learning*. A avaliação da linha de produtos desenvolvida foi feita no ambiente de uma fábrica de *software*, com foco em mensurar o *time-to-market* e quantidade de defeitos dos produtos gerados. Segundo o autor, os resultados mostram que a adoção da LPS é benéfica no que tange estas duas variáveis, evidenciando a aplicabilidade no mercado.

Já os serviços permitem a modelagem de componentes arquiteturais de um sistema como funcionalidades disponibilizadas remotamente (ROSEN *et al.*, 2008), o que possibilita que os usuários possam acessar as operações de um sistema remotamente, e que os proprietários possam atualizar, implantar e manter seus serviços sem que os usuários sejam privados de acessar a aplicação devido a um período de indisponibilidade. Neste modelo, os usuários notariam a indisponibilidade apenas das funcionalidades dependentes de serviços indisponíveis.

De acordo com Lee e Kotonya (2010), é possível unir as abordagens de LPS e orientação a serviços, criando o que os autores definem como Linha de Produtos Orientada a Serviços (LPOS). Esta abordagem apresenta o processo de mapeamento entre características e serviços, permitindo que uma linha de produtos de *software* tradicional possa ter seus *assets* mapeados para um conjunto de serviços, permitindo que instâncias daquela linha de produtos possam usufruir dos benefícios gerados pela orientação a serviços.

1.2 Problema em Destaque

Conforme mencionado anteriormente, o grau de retrabalho apresentado em estudos publicados no domínio de aplicações de *E-learning* é elevado. Para sanar este problema, se fazem necessárias pesquisas com o objetivo de explorar a aplicabilidade de técnicas de reúso de *software* neste domínio de aplicações.

Conforme o trabalho de Junior (2015), citado anteriormente como uma solução válida, é possível aplicar a técnica de linhas de produtos de *software* satisfatoriamente. No entanto, tal estudo estabelece, como limite de escopo, apenas a abrangência de sistemas *M-learning*, o que significa a incerteza sobre a aplicabilidade da técnica de LPS para sistemas *P-learning* e *U-learning*. Além disso, o autor também aplica a técnica de LPS, sem utilizar conceitos de orientação a serviços. Portanto, o estudo não investiga se a abordagem de LPOS é aplicável para sistemas de aprendizagem, sem validar se a orientação a serviços pode ser utilizada nesta categoria de sistemas computacionais.

Este trabalho foca suas questões de pesquisa no problema gerado pela união dos problemas citados: investigar se a técnica de modelagem de LPOS é aplicável no âmbito dos sistemas de aprendizagem, abrangendo os sistemas de tipo *U-learning*. Como os dispositivos IoT têm seu

uso potencializado pela orientação a serviços e objetivam, por definição, habilitar a ciência de contexto das aplicações onde são utilizados (ATZORI; IERA; MORABITO, 2010; AL-FUQAHA *et al.*, 2015), esta investigação deve explorar o uso de tais dispositivos, uma vez que sistemas *U-learning* exigem a presença de quaisquer traços de ciência de contexto por parte do sistema (HWANG; TSAI; YANG, 2008).

1.3 Solução Desenvolvida

Este trabalho apresenta a *Museum Touring and Tutoring Product Line* (MTTPL), uma LPOS desenvolvida para auxiliar a instanciação de novos sistemas de aprendizado, com foco em sistemas de auxílio à visitação de museus. Esta foi elaborada com base em um processo de engenharia de domínio, consultando especialistas em sistemas de aprendizagem para compreender melhor as necessidades de tais sistemas, bem como as restrições a serem obedecidas. Também foi tomado como base o Mapeamento Sistemático, ou *Systematic Mapping* (SM) realizado anteriormente (FUZETO; BRAGA, 2016), e que é apresentado em detalhes no Capítulo 3.

Esta LPOS apresenta um modelo de características, que mapeia seus pontos de comunidade e variabilidade, bem como um modelo de serviços, mapeando conjuntos de *features* para serviços distintos. Por fim, a proposta da MTTPL também apresenta modelos de classes para cada serviço, provendo *assets* que descrevem detalhes de implementação fundamentais para uma criação adequada de quaisquer instâncias da LPOS.

Para validar a proposta da MTTPL, também é apresentado neste trabalho o SmartICMC: um sistema criado a partir da LPOS proposta para auxiliar a navegação dos visitantes do ICMC no Museu da Fauna e da Flora. Para este sistema, são apresentados os artefatos específicos, instanciados a partir dos artefatos da MTTPL, bem como os diagramas de classe dos serviços instanciados e partes do código-fonte gerado para ilustrar o resultado final obtido.

Como resultado, foi avaliado que a MTTPL foi capaz gerar uma instância correta, capaz de gerar produtos que atendam à demanda do ambiente de operação. Este resultado valida a proposta da MTTPL, como uma LPOS válida para o domínio de aplicação abrangido por ela.

Segundo a questão de pesquisa levantada, os resultados obtidos a partir do SmartICMC geram respostas positivas, deixando claro que é possível aplicar a técnica de LPOS para sistemas de aprendizagem, com uma margem de segurança maior para sistemas de auxílio à visitação de museus.

1.4 Estrutura do Documento

No Capítulo 2, são apresentados os conceitos envolvidos na elaboração da MTTPL, tais como os fundamentos de IoT, sistemas de aprendizado, serviços e LPS. No Capítulo 3, é apresentado o mapeamento sistemático feito para identificar as formas com que tecnologias IoT

são aplicadas em ambientes E-learning, M-learning e U-learning. No Capítulo 4, é apresentado o *design* da MTTPL, partindo do resultado do SM, passando pelo diagrama de casos de uso e modelo conceitual, elaborando o modelo de *features*, até definir os serviços gerados pela MTTPL e os respectivos modelos de classe, além de realizar o mapeamento entre *features* e *assets*. No Capítulo 5, é apresentado o SmartICMC, sistema desenvolvido utilizando a modelagem da MTTPL como base e que exemplifica como o processo de desenvolvimento de novos sistemas usando a LPOS proposta deve ser conduzido. No Capítulo 6 são apresentados os resultados obtidos após a instanciação do SmartICMC, analisando os artefatos obtidos, verificando se os objetivos traçados para a MTTPL e o SmartICMC foram atingidos, e levantando e analisando as possíveis ameaças à validade deste trabalho. Por fim, no Capítulo 7 são apresentadas as conclusões acerca deste trabalho, discutindo brevemente os resultados que a MTTPL oferece à comunidade, bem como possíveis trabalhos futuros derivados da limitação de escopo deste trabalho.

FUNDAMENTAÇÃO TEÓRICA

Conforme citado anteriormente, esta proposta de trabalho envolve o uso de tecnologias IoT para *Learning Management Systems* (LMSs). No entanto, é necessário entender quais são as diversas propriedades destes conceitos, e quais tecnologias e paradigmas computacionais integram as discussões feitas ao redor destes conceitos. Por isso, neste capítulo são apresentados todos os conceitos necessários para o entendimento completo desta proposta de trabalho.

Este capítulo é organizado da seguinte forma: na [Seção 2.1](#), são apresentados os fundamentos de LMSs e a categorização dos mesmos segundo a literatura; na [Seção 2.2](#), são apresentados os conceitos de IoT, introduzindo as principais características dos dispositivos deste tipo e exemplos de aplicações que envolvam o uso destes; na [Seção 2.3](#), é apresentado o conceito de serviços de *software* e orientação a serviços, descrevendo as propriedades de um serviço, os contratos de uso e descrevendo o que são *web services*; na [Seção 2.4](#) é descrita a abordagem de LPS, descrevendo como se dá o mapeamento de pontos de variabilidade; por fim, na [Seção 2.5](#) é apresentada a técnica de LPOS, comentando o processo de mapeamento entre características e serviços, e comentando brevemente sobre LPOSs com reconfiguração dinâmica.

2.1 Learning Management Systems

Atualmente, as atividades de ensino acontecem, em sua maioria, nas salas de aula. São locais específicos onde um professor tem o trabalho de transmitir seu conhecimento a uma turma de alunos. Há séculos este tipo de ensino é aplicado em escolas, universidades, faculdades e outros tipos de instituições de ensino. No entanto, estudos recentes no campo da pedagogia questionam a eficiência deste modo de ensino, investigando a qualidade do ensino provido aos alunos, bem como sua motivação em continuar aprendendo ([MCPARLAND; NOBLE; LIVINGSTON, 2004](#); [HAKE, 1998](#)).

Ao mencionar as mudanças que ocorrem com os métodos de ensino e aprendizado, é

fundamental citar os avanços tecnológicos relacionados. Com a evolução dos computadores e o desenvolvimento de sistemas cada vez mais complexos, pesquisadores interessados na evolução da educação começam a idealizar novas formas de utilizar tais tecnologias para beneficiar o aprendizado dos alunos, através da formulação de aulas e módulos de ensino de maior qualidade, e da mudança dos ambientes de ensino visando um maior engajamento por parte dos alunos.

2.1.1 Tipos de sistemas de aprendizado

Um “*Learning Management System*” ou “Sistema de Gerenciamento de Aprendizado” (LMS) é um sistema computacional construído para dar suporte ao processo de ensino e aprendizado. Isto pode ocorrer através do gerenciamento dos materiais didáticos, disponibilização de tais materiais via *web*, estabelecimento de um canal de comunicação entre o professor e alunos, entre outras funcionalidades que possam aprimorar o ensino e aprendizado. No entanto, os diversos LMSs são divididos em categorias, que visam a classificação dos sistemas de acordo com as características das tecnologias presentes, que conferem aos LMSs diferentes potenciais de aplicação.

- **Aprendizado eletrônico (E-learning):** segundo [Maurer \(2003\)](#), sistemas E-learning devem permitir que os usuários possam se comunicar, gerenciar o aprendizado fornecido e adquirido, e anotar os materiais de ensino com informações relevantes. Desta forma, sistemas E-learning fornecem uma experiência de aprendizado personalizada. É importante lembrar que *todos* os sistemas computacionais de apoio ao ensino e aprendizado são classificados como sistemas E-learning.
- **Aprendizado móvel (M-learning):** de acordo com [Kearney et al. \(2012\)](#), um sistema M-learning aprimora o processo de aprendizado através do emprego de um dispositivo móvel. A aplicação deste tipo de dispositivo deve trazer algum benefício quanto à localização do usuário, indo além da simples utilização do sistema em qualquer lugar. Os benefícios considerados para um sistema M-learning, em sua maioria, são classificados como sociais, culturais e comunitários, indicando que o usuário deste tipo de sistema entra em contato direto com os objetos de aprendizado, assim enriquecendo o conhecimento adquirido.
- **Aprendizado pervasivo (P-learning):** este tipo de sistema adiciona o uso de tecnologia pervasiva, isto é, imperceptível pelos usuários. Quando a tecnologia não é visível, o usuário se vê em ambientes mais realistas, aumentando, assim, a motivação ([SHEN; WANG; SHEN, 2009](#)).
- **Aprendizado ubíquo (U-learning):** ao unir as tecnologias M-learning e P-learning, e adicionando a *ciência ao contexto* dos usuários, obtém-se um sistema U-learning. Este tipo de sistema se caracteriza pela possibilidade dos usuários aprenderem nos *lugares e momentos certos*. Para que isto ocorra apropriadamente, os dispositivos pervasivos

devem ser capazes de coletar dados sobre o usuário e seu atual estado. Este conjunto de informações é formalmente chamado de *contexto*. O sistema, ao interpretar o contexto do usuário, deve fornecer objetos de aprendizado que se adequem ao usuário no momento em que são fornecidos, assim criando uma experiência de aprendizado inteligente.

2.1.2 Contexto

Contexto é o conjunto de informações que definem o atual estado de um usuário. O contexto pode conter informações como localização, interesses, endereços de contato e objetos de aprendizado já utilizados, por exemplo, e não se limitando a estes (ABOWD *et al.*, 1999). Com as informações sobre o contexto de um usuário, o sistema pode auxiliar *ativamente* no aprendizado através da recomendação de objetos de aprendizado relacionados, de alguma forma, ao contexto. Quando um sistema se utiliza do contexto dos usuários e dispara ações através das mudanças de contexto, pode-se dizer que ele é *sensível ao contexto* (HWANG *et al.*, 2009).

Para que o contexto de um usuário seja coletado, um sistema educacional deve ser *ciente do contexto*. Esta característica diz respeito à capacidade do sistema de coletar as informações do contexto dos usuários que sejam relevantes às atividades suportadas pelo sistema. Para adquirir esta qualidade, o sistema deve ser capaz de se comunicar com dispositivos remotos ubíquos que possam coletar, de forma contínua, as informações do ambiente que o sistema deva usar. Os dispositivos usados para esta finalidade devem coletar informações da forma menos intrusiva possível, para que a experiência de aprendizado não seja influenciada negativamente. Por isto, dispositivos IoT são úteis para desempenhar esta função, uma vez que são ubíquos.

2.2 Internet das Coisas - IoT

Internet das Coisas é um paradigma computacional recente, tendo algumas das primeiras conferências sobre o assunto datando de meados de 2008. Presser *et al.* (2011) afirmam que começa-se a falar de IoT quando os dispositivos computacionais, sendo capazes de mensurar características do ambiente onde se encontram, começam a responder a certos eventos de forma autônoma e inteligente. Já Atzori, Iera e Morabito (2010) considera que IoT se trata da presença de dispositivos alocados pervasivamente pelo ambiente de inserção dos usuários, e que são capazes de cooperarem para alcançar objetivos em comum.

Este padrão de comportamento dos dispositivos IoT compreende alguns aspectos: a informação coletada por cada dispositivo é específica de um ambiente limitado, o que cria conjuntos de dados coletados em pequena granularidade. No entanto, a consolidação de todos os dados de todos os sensores permite a visualização da situação de uma área grande, desde uma visão mais abrangente até níveis de visualização mais próximos, com informações mais precisas de cada local. Esta precisão e flexibilidade dos dados coletados permite que o sistema forneça

informações mais compreensíveis sobre a situação dos ambientes monitorados, sem perder a precisão necessária para dar suporte às medidas tomadas em uma região específica.

2.2.1 Exemplos de aplicações de IoT

A principal característica de dispositivos IoT é a sua utilização para tornar um serviço mais inteligente, logo mais eficiente, ou possibilitar uma nova forma de usar tais dispositivos de maneira que o novo serviço também seja inteligente. No livro de [Presser et al. \(2011\)](#) são citados alguns exemplos:

- **Iluminação pública:** ao utilizar sensores de presença, o sistema é capaz de gerenciar quais pontos das ruas de uma cidade precisam, efetivamente, que a iluminação esteja funcionando. Adicionalmente, o sistema pode acender as luzes prevendo o trajeto dos cidadãos, evitando que um cidadão entre em um local escuro antes de iluminá-lo. Obviamente, esta funcionalidade do sistema visa a economia de energia elétrica, reduzindo os gastos públicos. No entanto, também é possível utilizar este sistema para destacar pontos onde os cidadãos devem tomar mais cuidado ao passarem, como locais de acidentes e buracos nas ruas, por exemplo.
- **Coleta de lixo:** quando são utilizados sensores para medir o peso de latas de lixo, um sistema de informação pode calcular quais rotas os caminhões de lixo de uma cidade devem percorrer num dado dia. Fazendo com que os caminhões de lixo percorram os menores trajetos possíveis, e permitindo que as lixeiras da cidade nunca fiquem repletas de lixo, o sistema é capaz de gerenciar as rotas de coleta de lixo de tal forma que o combustível necessário para os caminhões é reduzido, assim como o tempo necessário para realizar a coleta de lixo de toda uma cidade. Além disso, o sistema pode medir a produção de lixo de cada endereço, e o grau de lixo que é destinado à reciclagem, incentivando a conscientização dos cidadãos.
- **Planejamento:** medidores de poluição podem identificar, em tempo real, pontos de uma cidade onde há maior tráfego de veículos. Com isso, a administração pública pode planejar novos meios para redirecionar e reduzir a frota de veículos nas regiões de maior poluição, buscando assim melhorar a qualidade de vida das pessoas que habitam estes locais.
- **Saúde:** um médico ou enfermeira, munido de um dispositivo móvel com o software apropriado instalado, pode ler etiquetas de identificação de pacientes, remédios, e administrar a dosagem da medicação dos seus pacientes. Além disso, o sistema também pode registrar quais remédios um paciente tem comprado recentemente, bem como as visitas médicas feitas e os males diagnosticados por outros agentes de saúde. Desta forma, um agente pode coletar, de forma quase instantânea, todas as informações relevantes sobre um paciente e tomar as medidas apropriadas. Além disso, a etiquetagem de frascos de remédios pode

eliminar a necessidade de receitas transcritas em papel, podendo pacientes e agentes realizarem o pedido de mais remédios eletronicamente.

2.3 Serviços de Software

Serviços são “unidades discretas de negócio disponibilizadas por meio de um contrato de serviço” (ROSEN *et al.*, 2008). O contrato de serviço define *o que* o serviço faz, *quais parâmetros* são passados como entrada e saída de cada operação, e *como* tais funcionalidades são usadas e oferecidas. Desta forma, os consumidores podem ter acesso a um dado serviço com as garantias providas pelo fornecedor, sem terem conhecimento e/ou acesso à implementação do serviço utilizado. Isto significa que as tecnologias que um serviço utiliza internamente são irrelevantes para os consumidores. A Figura 1 apresenta a estrutura de um serviço.

Figura 1 – Estrutura de um serviço, adaptada de Rosen *et al.* (2008).

powered by Astah

Conforme mencionado, um serviço é oferecido aos consumidores através de um contrato de serviço. Na Figura 1, o contrato de serviço é representado pelo *Acordo de Nível de Serviço* e pelas *Políticas* definidos para ele. O Acordo de Nível de Serviço ou *Service Level Agreement* (SLA) estabelece os parâmetros de qualidade que o fornecedor se compromete a entregar, definindo um nível mínimo de performance em que os consumidores podem confiar. No entanto, os parâmetros de qualidade definidos no acordo valem para as *Políticas* de uso do serviço, às quais os consumidores devem observar e obedecer ao utilizar o serviço oferecido. Desta forma, garante-se que o fornecedor está entregando um serviço com o comprometimento de um certo nível mínimo de qualidade, e que os consumidores estão cientes de como utilizar aquele serviço da maneira correta.

Quanto às estruturas internas, um serviço é composto por:

- Interface: estruturas que lidam com a "visibilidade" do serviço, permitindo que aplicações externas possam acessar as funcionalidades implementadas internamente ao serviço, desde que através das APIs definidas. As estruturas de interface são divididas entre:
 - Operações: são os padrões para as funcionalidade do serviço. As operações de um serviço definem o que ele faz, quais parâmetros de entrada e saída são envolvidos em cada cenário de uso, e os protocolos de utilização e disponibilização das funcionalidades descritas.
 - Objetos semânticos: são as abstrações dos dados referentes ao modelo de negócio em que o serviço é ofertado pelo proprietário. Em geral, estes objetos são referências aos modelos e esquemas de dados de negócio.
- Implementação: estruturas que implementam, efetivamente, o serviço. São divididas entre:
 - Funcionalidades: são os métodos e funções implementados para realizar as operações relativas a cada cenário de utilização do serviço.
 - Dados: são as estruturas de dados, bem como métodos e tecnologias de armazenamento de informação utilizadas pelas funcionalidades do serviço para lidar com as informações recebidas, geradas e fornecidas.

2.3.1 Propriedades de serviços

Segundo Rosen *et. al.* (ROSEN *et al.*, 2008), um serviço com boa qualidade deve possuir as seguintes propriedades:

- Modularidade e granularidade: serviços são projetados como módulos bem-definidos, fáceis de serem arranjados dentro de uma *Service-Oriented Architecture* (SOA). Além disso, o nível de granularidade define o nível de riqueza que pode-se encontrar dentro de um serviço. Enquanto serviços de maior granularidade oferecem operações de mais alto nível, complexas e mais significativas, os serviços de menor granularidade oferecem operações de mais baixo nível, simples e fáceis de serem construídas.
- Encapsulamento: um serviço deve ter uma separação bem clara das estruturas de *implementação* da *interface*. Isto faz com que as estruturas de implementação sejam invisíveis aos consumidores, tendo de acessá-las através das interfaces do serviço.
- Acoplamento fraco: serviços devem possuir poucas dependências, que são bem conhecidas e gerenciadas. Manter o acoplamento de um serviço fraco permite que ele seja flexível e extensível, enquanto permite um melhor gerenciamento das dependências necessárias.

- Isolamento de responsabilidades: um serviço deve ser responsável por um conjunto de operações e informações específicas. Isto permite que um serviço contenha dados e funcionalidades específicas e bem definidas, reduzindo redundâncias.
- Autonomia: serviços devem ser implantados, modificados e mantidos independentemente de outros serviços e sistemas.
- Reúso: as características acima permitem que um serviço possa ser utilizado por diversos consumidores de diversas localidades e em múltiplos contextos. Pode-se dizer que serviços são utilizados como “tijolos” na construção de processos e serviços compostos.
- Descoberta e Associação Dinâmica: serviços podem ser descobertos através de um repositório de serviços. De acordo com (VERBORGH *et al.*, 2011), é possível a descoberta de novos serviços em tempo de execução. De qualquer forma, um consumidor deve poder ser associado dinamicamente ao serviço apropriado.
- *Stateless*: serviços não armazenam o estado da operação realizada pelos usuários. Isso significa que, para cada operação do serviço, não é interessante saber quais operações foram realizadas antes e quais podem ser as próximas. Serviços também devem ser independentes do estado e contexto de outros serviços, sendo dependentes apenas de suas funcionalidades.
- Auto-descritivo: os contratos de operação de um serviço devem descrever todas as informações relevantes aos consumidores: quais são as operações oferecidas, as interfaces de acesso, e os parâmetros de entrada e saída envolvidos. Os contratos também podem conter pré e pós-condições, bem como restrições quanto às operações.
- Composto: serviços podem ser compostos de outros serviços. Por consequência, um serviço pode ser combinado com outros a fim de criar um novo serviço.
- Governado por políticas: as relações entre consumidores e fornecedores devem ser governadas por políticas e SLAs.
- Independente de local, linguagem e protocolo: serviços devem ser acessíveis para qualquer usuário autorizado, em qualquer lugar e qualquer plataforma.

2.3.2 Tipos de serviço

Uma forma de categorizar diferentes serviços através de tipos é feita pelo tamanho, escopo e propósito. Esta diferenciação permite que o projeto arquitetural do sistema seja dividido em *camadas*, identificando quais tipos de serviços serão necessários em cada camada (ROSEN *et al.*, 2008). Assim, sabe-se, a cada camada arquitetural, qual o nível de complexidade e o tipo de operações que um dado serviço terá.

Uma das classificações típicas para serviços divide-os em três camadas diferentes, de acordo com as regras de gerenciamento de negócio (ou *Business Management Rules* - BMP):

- **Serviços operacionais:** são os serviços que implementam as operações de negócio, como, por exemplo, validar um número de telefone de acordo com o país ou calcular os juros devidos por clientes. Serviços operacionais podem ter diferentes dimensões, podendo variar de serviços simples e com propósitos muito bem definidos e restritos, até serviços grandes, complexos e com grande valor significativo para os negócios do fornecedor.
- **Serviços de entidade:** são os serviços que fornecem acesso às entidades de um negócio, como clientes, funcionários, contas e históricos de pagamento. Estes serviços, usualmente, têm tamanho médio a grande. Também costumam ser independentes de todos os processos de negócio, no entanto fazendo parte de todos eles. Importante notar, também, que como estes serviços fornecem apenas o acesso às entidades de negócio, apresentam alto grau de reuso. Também é importante notar que entidades não são esquemas de dados e/ou elementos dentro de um esquema.
- **Serviços de decisão:** são os serviços que executam as regras de negócio, a fim de encontrar as decisões mais adequadas. Um exemplo de serviço de decisão é “Aprovar aumento de salário”, que pode responder sim ou não de acordo com o registro de um funcionário, constante no sistema. Um serviço de decisão geralmente fornece respostas sim ou não. No entanto, também podem prover suporte para operações que façam uso de regras externas, como taxas governamentais em constante mudança (inflação por exemplo). Serviços de decisão normalmente são compostos por outros serviços, que variam de pequeno a médio porte.

2.3.3 *Web services*

Para que as interfaces bem definidas de um serviço possam ser utilizadas, elas devem ser de fácil acesso. Da mesma forma, a tecnologia utilizada para disponibilizar um serviço também deve permitir que ele seja de fácil manutenção e implantação. A saída comumente utilizada, conferindo melhores resultados nestes aspectos, é a implementação de serviços através de um *web service*.

Web services são componentes de software autônomos, capazes de serem descritos, publicados, descobertos e invocados utilizando um conjunto de tecnologias *web* conhecidas (CHUKMOL; BENHARKAT; AMGHAR, 2009). Algumas destas tecnologias são *Simple Object Access Protocol* (SOAP), *Web Services Description Language* (WSDL), *Uniform Resource Identifier* (URI) e *Web Application Description Language* (WADL) (RICHARDSON; RUBY, 2008).

Richardson e Ruby (2008) também afirmam que “toda aplicação *web*- toda página *web*- é um serviço”. Assim, é possível visualizar as URLs de uma página *web* como as interfaces de um serviço, o que facilita o entendimento de como um serviço pode ser disponibilizado e utilizado, através de um *web service*.

Um exemplo de como um *web service* se encaixa nestas qualidades é a página de buscas da Google¹. Ao clicar no botão “Pesquisa Google”, a página envia um pedido, para o servidor, para que a busca do usuário seja feita, utilizando a *string* do campo de entrada como chave. Como resposta, é exibida a primeira página dos resultados da busca efetuada. Ao visualizar esta operação como parte de um serviço, pode-se idealizar a página inicial como uma resposta de uma operação, invocando a página inicial do motor de busca. Além disso, o clique no botão “Pesquisa Google” também pode ser visto como o disparo de outra operação deste serviço, acessando a interface do motor de busca Google que realiza uma busca pela *string* informada. Portanto, é possível ver que as operações do motor de busca são acessíveis via uma interface bem definida: a URL, que incorpora os parâmetros da busca realizada. Por isso classifica-se toda a parte de busca da Google como um *web service*.

2.3.3.1 SOAP

O SOAP é utilizado em cima do HTTP, sendo um formato de envelopamento de requisições *web* baseado em XML (do inglês *eXtensive Markup Language*) (RICHARDSON; RUBY, 2008). A princípio, o SOAP encapsula a mensagem HTTP dentro de um documento XML com a estrutura de um *Remote Procedure Call* (RPC). Isto pode parecer controverso ao lembrar que requisições *web* são feitas através de *mensagens*, geralmente HTTP. No entanto, a estrutura do SOAP é comumente utilizada desta forma, mesmo que o protocolo trate apenas do envelopamento das mensagens HTTP.

Em termos de estrutura, o envelope SOAP não é muito diferente do HTTP, tendo os corpos e cabeçalhos semelhantes e com os mesmos objetivos. A maior diferença entre os dois é a independência do SOAP do meio de transporte. O SOAP envelopa, também, o cabeçalho, deixando-o dentro da mensagem. Assim, o destinatário e intermediários não precisam ser capazes de ler e entender a mensagem, mas sim de transmiti-la e recebê-la. Apenas o destinatário precisa ter a tecnologia necessária para abrir o envelope SOAP e traduzir as informações contidas nele.

2.3.3.2 REST

O *REpresentational State Transfer* (REST) é uma tecnologia originalmente criada para a construção de sistemas hipermídia de larga escala (PAUTASSO; ZIMMERMANN; LEYMANN, 2008). É comum encontrar o uso do termo REST em conjunto com HTTP, uma vez que o padrão REST é bastante utilizado para *web services* e já é sabido como o REST pode contribuir para a escalabilidade de sistemas com interfaces HTTP.

O REST é baseado em quatro princípios:

- Os recursos de um *web service* RESTful devem ser disponibilizados de forma que sejam facilmente identificáveis e utilizáveis. A identificação destes recursos é feita através da

¹ <https://www.google.com.br/>

URI.

- Os recursos supracitados são acessados e utilizados via um conjunto de operações *Create*, *Read*, *Update* e *Delete* (CRUD). No caso de *web services* RESTful, que utilizam basicamente o protocolo de mensagens HTTP, estas operações são conhecidas como os comandos PUT, GET, POST e DELETE.
- As mensagens são auto-descritivas. Isto significa que uma mensagem pode ser lida independentemente de seu formato. Ou seja: a mesma mensagem pode ser interpretada como uma mensagem HTTP, um envelope XML, um documento PDF, uma imagem, ou simplesmente texto puro.
- Um *web service* RESTful é, internamente, *stateless*. Isto significa que o próprio sistema não armazena informações sobre o contexto de execução das operações dos clientes, e não é do interesse do sistema armazenar tal informação. No entanto, é possível utilizar algumas técnicas para armazenar e identificar transições de estados, como, por exemplo, *caching* dos dados de uma página *web*.

Segundo [Pautasso, Zimmermann e Leymann \(2008\)](#), *web services* RESTful são adequados para os sistemas em que não há grande preocupação em garantir as características de transação, confiabilidade e segurança em nível de mensagem. Além disso, *web services* RESTful conseguem ser superiores aos baseados em SOAP, quando avaliados em termos de flexibilidade e controle. No entanto, os serviços baseados em SOAP não necessitam de uma codificação essencialmente em baixo nível.

2.4 Linhas de Produtos de Software

“Uma LPS consiste em uma família de sistemas de software que possuem funcionalidades comuns e algumas funcionalidades variáveis” ([GOMAA, 2005](#)). O desenvolvimento de LPSs visa o reúso de software, uma vez que diferentes produtos, gerados a partir da mesma LPS, reutilizam as funcionalidades comuns, ao menos.

Uma LPS, para atingir os níveis desejados de reúso de software, utiliza dos conceitos de *comunalidade* e *variabilidade*. A comunalidade de uma LPS é descrita de acordo com os módulos comuns a todos os produtos, enquanto a variabilidade é descrita pelos módulos opcionais ou variantes, isto é, os módulos que aparecem em pelo menos um produto, mas não em todos.

Uma LPS possui todos os artefatos presentes no ciclo de vida de um software qualquer, claramente com alguns adicionais oriundos da modelagem das variabilidades entre produtos. Por isso, há uma série de custos e benefícios relativos à modelagem e definição de cada artefato. No entanto, a relação custo-benefício mais observada no desenvolvimento de LPSs é a de *tempo*. Enquanto uma LPS leva mais tempo do que um software tradicional para ser desenvolvida,

quando pronta é capaz de instanciar quaisquer produtos rapidamente, acelerando o processo de desenvolvimento para *todos* os produtos compreendidos pela LPS. É de suma importância observar que o custo cresce ainda mais caso a LPS seja mal projetada e/ou desenvolvida, e demande constante correção e manutenção.

Para desenvolver uma LPS, é necessário, em um primeiro momento, definir quais produtos serão utilizados para definir os componentes comuns e variáveis. Assim, são definidas duas abordagens diferentes (BACHMANN *et al.*, 2012):

- Proativa: esta abordagem foca em não basear-se em softwares legado, mas realizar análise de domínio e definir o escopo dos produtos a serem produzidos. Assim, é possível definir as limitações e funcionalidades de todos os produtos desejáveis da LPS, o que permite definir quais serão os componentes comuns e quais serão variáveis.
- Reativa: ao contrário da abordagem proativa, esta toma como base um conjunto de produtos sobre os quais deseja-se construir uma LPS. Assim, cada produto analisado gera um conjunto de componentes, e a comparação entre os conjuntos permite identificar os pontos de comunalidade e variabilidade.

2.4.1 Variabilidade

As variabilidades de uma LPS são medidas por seus componentes, classificados como variáveis. Um componente variável aparece, ao menos, em um produto, porém não em todos.

Segundo Gomaa (2005), os componentes variáveis se dividem entre componentes opcionais, que aparecem em alguns dos produtos da LPS, e componentes alternativos, que representam um conjunto essencialmente mutuamente exclusivo. Enquanto um produto pode conter todos os componentes opcionais, cada produto pode conter apenas um dos componentes alternativos pertencentes ao mesmo grupo. Outra observação a ser feita é que um grupo de componentes alternativos pode ser comum, isto é, cada produto da LPS contém um dos componentes deste grupo. Apesar disto, um grupo de componentes alternativos também pode ser opcional, no entanto esta característica deve ser explicitada durante o planejamento da LPS.

A modelagem de pontos de variabilidade para LPSs é feita de diversas formas, como a abordagem *Family-Oriented Abstraction, Specification and Translation* (FAST) (COPLIEN; HOFFMAN; WEISS, 1998) e a *Evolutionary Domain Life Cycle* (EDLC) (GOMAA; FARRUKH, 1998). No entanto, a modelagem de *features* é a mais utilizada para modelar variabilidades em requisitos (GOMAA, 2005).

2.4.2 Features

Uma *feature* é uma característica presente em produtos de uma LPS. As *features* representam as características reutilizáveis de uma linha de produtos, e podem ser utilizadas para

diferenciar membros da linha de produtos, diferenciando, assim, características comuns de variáveis (GOMAA, 2005).

O processo de análise de uma LPS é sumariamente importante, dado que métodos como o *Feature-Oriented Domain Analysis* (FODA) (KANG *et al.*, 1990) dependem da análise dos requisitos de uma linha de produtos, a fim de identificar os pontos de comunalidade e variabilidade da LPS. Como *features* podem apresentar comportamentos variados e complexos, a atividade de análise de *features* é mais complexa para LPSs do que para sistemas únicos. A Figura 2 mostra as três atividades essenciais para o desenvolvimento de uma linha de produtos, segundo Clements e Northrop (2002), e representa a importância da análise para o ciclo de vida de uma LPS, através da sua influência em todas as atividades do seu ciclo de vida.

Figura 2 – As três atividades principais do ciclo de vida de uma LPS, segundo Clements e Northrop (2002).

Segundo Gomaa (2005), as *features* podem ser categorizadas da seguinte forma:

- Comuns: são as *features* que aparecem em todos os produtos da LPS. *Features* comuns sempre serão incluídas na instanciação de um produto.
- Opcionais: são as *features* que aparecem apenas em alguns dos produtos. Podem ser incluídas durante a instanciação de um produto, porém apenas adicionando uma característica, e não uma funcionalidade essencial. Um produto pode ser instanciado sem nenhuma *feature* opcional.
- Alternativas: estas *features* são modeladas como conjuntos de *features* alternativas umas às outras. Isto significa que durante a instanciação de um produto, *features* alternativas são

mutuamente exclusivas, isto é, a inclusão de uma automaticamente exclui as demais. Uma LPS pode conter zero ou mais grupos de *features* alternativas.

- Parametrizadas: são *features* que definem parâmetros dos produtos instanciados. Uma *feature* parametrizada demanda, no momento da configuração do produto, que o valor do parâmetro seja definido. Tal *feature* deve especificar o tipo do parâmetro, o intervalo ou valores permitidos, e opcionalmente um valor padrão.
- Pré-requisito: uma *feature* pode ser modelada como pré-requisito de uma ou mais outras *features* da linha de produtos. Este comportamento define que a *feature* pré-requisito deve ser instanciada, para que as dependentes possam ser instanciadas. A definição deste comportamento elimina problemas que a ausência de uma *feature* poderia causar a outras que dependam desta.
- Mutuamente inclusivas: estas *features* se comportam de maneira semelhante às alternativas, porém com lógica inversa: em um grupo de *features* mutuamente inclusivas, a inclusão de uma destas causa, necessariamente, a inclusão de todas as outras mutuamente inclusivas à *feature* incluída inicialmente.

A modelagem de *features* pode ser feita de diversas formas. Para quaisquer tipos de sistema, é possível modelá-las através de uma tabela, listando cada *feature*, sua classificação e o ponto correspondente de variabilidade. Para sistemas OO, no entanto, a abordagem mais utilizada se concentra na análise de variabilidade do diagrama de casos de uso. Gomaa (2005) apresenta o processo de análise baseado nos casos de uso, em que grupos de casos de uso classificados de forma semelhante podem ser empacotados dentro de uma mesma *feature*, através da utilização da representação de pacotes, fornecida pela Unified Modeling Language (UML). A Figura 3 apresenta um exemplo de como as *features* de uma LPS podem ser modeladas.

2.4.3 Product Line UML-based Software Engineering

O método *Product Line UML-based Software Engineering* (PLUS) foi desenvolvido por Gomaa (2005), com o intuito de estender o uso da UML para LPS, linguagem que até então era utilizada apenas para descrever sistemas únicos. Segundo o autor, o maior benefício no uso da PLUS, além da utilização da UML como linguagem de descrição, é a modelagem explícita das comunicações e variabilidades de uma linha de produtos de *software*.

A PLUS baseia-se, primordialmente, na utilização de *diagramas de caso de uso* e *modelos de features*. Apesar destes dois diagramas serem essências para a descrição de uma LPS segundo este método, é possível utilizar uma abordagem mais formal, segundo o autor, utilizando-se de outros modelos estáticos, modelos dinâmicos e até *statecharts*.

Segundo Gomaa (2005), o PLUS é utilizado pelo processo conhecido por *Evolutionary Software Product Line Engineering Process* (ESPLEP), que “elimina a distinção tradicional

Figura 3 – Exemplo de modelo de *features*.

entre desenvolvimento e manutenção de *software*”. Este processo introduz dois ciclos de vida importantes para uma LPS que seja desenvolvida sob este processo, os quais são descritos na subseção a seguir. A Figura 4 apresenta a organização das etapas do ESPLEP.

Figura 4 – Engenharia de LPS usando o ESPLEP (GOMAA, 2005) (fonte: (DONEGAN, 2008)).

2.4.4 Engenharia de Aplicação e Engenharia de Domínio

Segundo Gomaa (2005), o *domínio de aplicação* de uma LPS consiste na arquitetura da mesma, e o processo de idealização deste modelo é comumente referenciado por *análise de domínio*. Assim, o método PLUS utiliza tal análise para modelar as variabilidades da LPS e definir sua família de produtos. A este processo, deu-se o nome *engenharia de domínio*.

Durante a *engenharia de aplicação*, “a arquitetura da linha de produtos de *software* é adaptada e desenvolvida para derivar uma dada aplicação de *software*, que seja produto da linha de produtos” (GOMAA, 2005). Desta forma, é esperado que, ao final da engenharia de aplicação, seja obtido um produto derivado da LPS que atenda aos requisitos desejados, através da instanciação *apenas* das características desejadas.

Ainda segundo o autor, a engenharia de aplicação deve seguir uma ordem de passos, definida e ordenada da seguinte forma:

1. **Modelagem de requisitos:** os requisitos da aplicação a ser gerada são utilizados para gerar modelos descritivos (e.g. diagrama de casos de uso), e uma análise comparativa com os modelos descritivos da LPS é feita, visando selecionar os elementos a serem instanciados;
2. **Modelagem da análise:** modelos descritivos de menor grau de abstração são elaborados, e quaisquer elementos menores necessários para descrever adequadamente as funcionalidades da aplicação são adicionados. Nesta fase, são citados modelos estáticos, como diagramas de classes, dinâmicos, como diagramas de comunicação;
3. **Modelagem de design:** os modelos derivados anteriormente são utilizados para derivar a arquitetura da aplicação gerada, a partir da arquitetura da LPS;
4. **Implementação incremental:** a aplicação é finalmente implementada. Nesta etapa, é desejado que o produto gerado demande um esforço mínimo de implementação, dado que, em se tratando de uma linha de produtos, hajam *assets* reusáveis. No entanto, nas fases iniciais de vida da LPS, a implementação incremental de um novo produto pode significar a implementação de todo o código-fonte ou de parte significativa do mesmo;
5. **Teste:** a aplicação gerada é testada, verificando se cada um dos incrementos incluídos a partir da LPS foram instanciados corretamente. É importante destacar que os incrementos não são testados apenas isoladamente, pois também devem ser realizados testes de integração entre tais incrementos, misturando testes de *caixa-preta* e *caixa-branca*.

2.5 LPS x SOA

Segundo Lee, Muthig e Naab (2008), a análise e adaptação de uma LPS para orientação a serviços é um método que visa a elaboração de serviços que atendam aos requisitos específicos

de cada produto da família considerada, bem como a evolução da linha de produtos através da incorporação das características de sistemas orientados a serviços, tais como flexibilidade e escalabilidade. Isto cria o que pode-se chamar de “Linha de Produtos Orientada a Serviços (LPOS)”.

Assim como o PLUS, este método propõe que a análise de uma linha de produtos seja baseada num conjunto de *features*. Após a obtenção do modelo que mapeia as variabilidades da LPOS, cada grupo de características é analisado separadamente, visando entender a força de sua relação com cada *feature* que o compõe. Nos casos em que a relação é forte o suficiente para a característica ter sentido existencial apenas dentro daquele grupo, ambos são mapeados para um mesmo serviço. Nos casos em que este vínculo não é tão forte, tal *feature* é passível de ser mapeada para um serviço distinto. Ainda assim, o resultado pode ser influenciado por questões mais subjetivas, como a necessidade aparente de isolar tal característica em um serviço à parte ou o “parecer” do engenheiro, no momento de tomar esta decisão.

Também é notável o fato desta análise ser aplicável retroativamente: com a definição de um serviço a partir de um conjunto de *features*, é possível que a análise daquele serviço a respeito do grupo em que está inserido resulte na criação de um novo serviço. Desta forma, o novo serviço pode incorporar diretamente as funcionalidades do serviço definido anteriormente, ou pode compor suas funcionalidades em função de outro serviço. Isto gera situações onde há a *composição* de serviços.

Assim, a análise de serviços da LPS pode reorganizar o modelo de *features* para agrupar as diferentes características da linha de produtos considerando duas classificações distintas de serviços: os serviços de orquestração, que tratam da coordenação das atividades de outros serviços a fim de cumprir com uma ou mais funcionalidades, e os serviços moleculares, que oferecem operações mais simples e servem como base para a composição de serviços mais complexos. Enquanto os serviços de orquestração devem ser construídos tomando cuidado com a correteza de seus comportamentos, os serviços moleculares devem ser definidos com níveis de granularidade corretos, visando graus de reusabilidade adequados.

2.5.1 Diferenças entre LPS, SOA e LPOS

Lee e Kotonya (2010) identificam uma diferença notável entre serviços e *features*. As características de capacidade de uma LPS (serviços, operações e características não-funcionais) são similares a serviços em uma SOA. No entanto, devido à obediência estrita aos requisitos de qualidade, sistemas modelados utilizando os conceitos de SOAs não comportam o mapeamento entre variabilidades e produtos com facilidade, o que dificulta a alteração de um sistema para gerar produtos distintos.

A Qualidade de Serviço, ou *Quality of Service* (QoS) é uma característica essencial em sistemas orientados a serviços, uma vez que a provisão de operações deve ocorrer desejavelmente

de forma fluida. Os níveis de qualidade que os fornecedores devem garantir estão descritos no SLA, e devem ser respeitados para que o contrato seja respeitado. No entanto, em uma LPS as características de qualidade são tratadas estaticamente, o que significa que a provisão de recursos do produto final pode não ser suficiente, assim deteriorando a QoS.

Quanto à orquestração dinâmica de serviços, é uma característica comum em serviços, especialmente quando é utilizado um padrão de descrição de serviços bastante difundido, tal como WSDLs. Aos olhos das LPSs, as características dos produtos são definidas previamente à implantação do produto final, e não consideram a composição dinâmica de *features*. Mesmo havendo a distinção entre serviços configurados estaticamente e dinamicamente, é bem provável que os serviços dinâmicos dependam de serviços de terceiros, o que pode oferecer risco às restrições dos serviços estáticos.

Já a LPOS visa compreender as características destas duas abordagens. Como esta abordagem visa gerar linhas de produtos de *software*, beneficia-se do mapeamento de variabilidades e na facilitação em instanciar diferentes produtos através da instanciação de conjuntos diferentes de características. Além disso, uma LPOS também colhe frutos por incorporar conceitos de SOAs, como a capacidade de prover funcionalidades com melhores medidas de QoS, bem como a possibilidade de implementar mecanismos de orquestração de serviços que utilizem repositórios dinâmicos de *web services*.

2.6 Considerações Finais

Este capítulo apresentou os principais conceitos envolvidos neste trabalho. Foram apresentados os LMSs, bem como a categorização e características; os principais conceitos de IoT, bem como exemplos de aplicações; os principais conceitos sobre serviços de *software*, bem como o que são SLAs, os diferentes tipos de serviços e o que são *web services*; os conceitos de linhas de produtos de *software*, descrevendo como é feito o mapeamento das variabilidades e como são gerados os *assets* de uma LPS; e, por fim, o que são as linhas de produtos orientadas a serviços, mostrando como LPSs e serviços são usados em conjunto na definição de novas linhas de produtos. Tais conceitos são fundamentais para embasar teoricamente as atividades apresentadas neste trabalho, bem como para compreender as decisões tomadas e os resultados obtidos.

No capítulo a seguir, é apresentado o mapeamento sistemático feito para delinear o estado-da-arte para este trabalho, visando entender quais lacunas ainda não foram preenchidas e que podem ser exploradas.

MAPEAMENTO SISTEMÁTICO

Para identificar o estado-da-arte da aplicação dos conceitos de IoT no domínio da educação de maneira adequada e suficientemente abrangente, foi necessário utilizar uma abordagem imparcial, que permitisse aos pesquisadores eliminar quaisquer vieses oriundos de olhares subjetivos aos trabalhos avaliados.

Considerando tal necessidade, optou-se por realizar um Mapeamento Sistemático, ou *Systematic Mapping* (SM). Tal abordagem, proposta por [Kitchenham et al. \(2007\)](#), baseia-se nos conceitos de uma Revisão Sistemática aplicada para a Engenharia de Software. Segundo Kitchenham et. al., uma Revisão Sistemática (SLR) é “um tipo de estudo secundário que usa uma metodologia bem-definida para identificar, analisar e interpretar todas as evidências relacionadas a uma questão de pesquisa, de forma que seja repetitível e livre de qualquer viés”. Ao tratar de SMs, os autores o definem como um segundo tipo de estudo secundário, com o objetivo de apresentar as evidências de um domínio de estudo em alto nível de granularidade, agrupando-as em áreas de similaridade e apontando linhas de pesquisa que demandem mais trabalhos e até mesmo SLRs. Complementando esta definição, [Petersen et al. \(2008\)](#) afirmam que a técnica de SM é recomendada quando não há um volume satisfatório de estudos primários relevantes e de alta qualidade. Além disso, a recomendação do uso de SM se estende aos casos em que hajam poucas evidências a serem coletadas, ou quando o tópico de pesquisa é muito abrangente.

Desta forma, ao utilizar o SM como abordagem de revisão bibliográfica, foram eliminadas ameaças à validade acerca de trabalhos relacionados através da aplicação de uma técnica estruturada e bem definida. Esta utilização também contribui com uma segurança satisfatória em relação à coleta de todos os trabalhos relevantes para o tema pesquisado. Um artigo foi publicado sobre o SM apresentado neste capítulo em um workshop de IoT em aplicações educacionais, mostrando resultados parciais ([FUZETO; BRAGA, 2016](#)).

É importante ressaltar que este SM já está consideravelmente desatualizado, dado que os estudos avaliados foram captados na segunda metade de 2016. Portanto, conforme mencionado

no [Capítulo 7](#), é vital que o mesmo seja atualizado futuramente, principalmente visando avaliar se os mesmos dados estatísticos apresentados neste Capítulo se mantêm verdadeiros.

3.1 Planejamento

Antes de conduzir este SM, foi necessário definir um protocolo, contendo informações sobre os objetivos do mapeamento, as bases de dados e *strings* de pesquisa a serem utilizadas, e os critérios de inclusão e exclusão de estudos. Nesta seção, estão descritas tais informações para que seja possível a replicação deste SM por outros pesquisadores. É importante notar que o protocolo aqui apresentado se refere à condução de um SM finalizada no início de 2017.

3.1.1 Objetivos e Questões de Pesquisa

O objetivo deste mapeamento é identificar os trabalhos publicados que relacionam o domínio da educação, aplicada em sistemas computacionais de apoio ao processo de ensino e aprendizagem, com o paradigma computacional da IoT. Os resultados obtidos permitem mapear as contribuições feitas para este domínio, identificando as linhas de pesquisa ativas e problemas a serem resolvidos. Para guiar o mapeamento, em um primeiro momento foram definidas as questões de pesquisa, que ajudaram a nortear a seleção dos estudos para aqueles de maior relevância, bem como os dados a serem coletados de cada estudo avaliado. Uma questão de pesquisa primária foi definida, que é a seguinte:

Questão de Pesquisa Primária: Quais estudos publicados relacionam a Internet das Coisas com a educação eletrônica (*E-learning*)?

Com base na Questão de Pesquisa Primária, foram derivadas questões de pesquisa secundárias, visando complementar o objetivo do mapeamento por meio da definição de dados considerados importantes para serem coletados dos estudos avaliados. São as questões secundárias:

- **Questão de Pesquisa Secundária 1:** Quais foram os tipos de sistemas de ensino pesquisados, estudados ou propostos (*E-learning*, *M-learning* e *U-learning*)?
- **Questão de Pesquisa Secundária 2:** Em quais anos houve maior volume de estudos publicados?
- **Questão de Pesquisa Secundária 3:** Quais são os países que contribuem mais para este domínio?
- **Questão de Pesquisa Secundária 4:** Quais são os tipos de contribuições publicadas?

- **Questão de Pesquisa Secundária 5:** Em quais campos da ciência foi investigada a aplicação de IoT na educação eletrônica?
- **Questão de Pesquisa Secundária 6:** Quais foram os recursos educacionais utilizados?
- **Questão de Pesquisa Secundária 7:** Quais foram os níveis de escolaridade considerados? Houve algum nível que nunca foi investigado?
- **Questão de Pesquisa Secundária 8:** Quais foram as tecnologias IoT utilizadas?
- **Questão de Pesquisa Secundária 9:** Os autores dos trabalhos publicados citam claramente o uso de IoT? Ou a utilização deste recurso é deixada implícita?

Apesar de seguirem a ideia de serem bastante claras e explícitas sobre sua relação com os objetivos do mapeamento, algumas questões de pesquisa secundárias incorporam observações feitas durante a fase de condução, demandando, assim, explicações mais detalhadas. As questões 8 e 9 foram definidas para facilitar a identificação das tecnologias utilizadas em cada estudo selecionado, a fim de identificar quais foram mais utilizadas e, potencialmente, tornaram-se tendência entre a comunidade científica. No entanto, durante os ciclos de refinamento da *string* de pesquisa, identificou-se que poucos estudos eram encontrados nas bases de dados selecionadas utilizando apenas os termos de Internet das Coisas e *Internet of Things*, no que se refere às tecnologias que desejava-se encontrar aplicadas na educação eletrônica. Para recuperar artigos relevantes, optou-se por eliminar a necessidade de encontrar tais termos explícitos no texto dos estudos, buscando também, então, pelas *tecnologias* conhecidas como relacionadas a sistemas IoT. Desta forma, a Questão de Pesquisa 9 tornou-se necessária para identificar os estudos que são selecionados pelas tecnologias pesquisadas, e não pelo paradigma computacional desejado.

A Questão de Pesquisa Secundária 1 trata da identificação de diferentes níveis de especialização de sistemas educacionais. Apesar deste mapeamento buscar identificar o uso da IoT no contexto do *E-learning*, cada um dos tipos de sistema identificados trata da incorporação de um novo aspecto que aproximam sistemas *E-learning*, em nível conceitual, de um cenário de aplicação em que dispositivos IoT possam ter um impacto positivo maior.

3.1.2 *String* genérica e dados coletados

Com as Questões de Pesquisa definidas, pode-se planejar e ajustar a *string* de pesquisa genérica, uma vez que já é sabido o que se deseja encontrar nas bases de dados a serem consultadas. Com base nas Questões de Pesquisa apresentadas anteriormente, obteve-se os termos e sinônimos a serem utilizados nas consultas, constantes na Tabela 1. A partir destes termos e sinônimos, foi derivada a seguinte *string* de pesquisa genérica:

(internet of things OR iot OR rfid OR wsn OR wireless sensor network OR sensor) AND
(e-learning OR ubiquitous learning OR u-learning)

Tabela 1 – Termos e sinônimos

Termo	Sinônimos
Internet of Things	IoT, RFID, WSN, Wireless Sensor Network, sensor
E-learning	ubiquitous learning, U-learning

Fonte: Dados da pesquisa.

Para a extração adequada dos dados dos artigos recuperados, foram definidos quais dados seriam extraídos, os quais foram descritos utilizando um formulário. Todos os formulários de extração de dados foram feitos utilizando a ferramenta Google Forms, com os resultados armazenados em uma planilha de dados Google Spreadsheet. A Tabela 2 apresenta cada um dos dados coletados, e sua respectiva relevância de análise.

Tabela 2 – Dados coletados dos estudos

Dado coletado	Relevância de análise
Ano	Períodos de pesquisa intensa, atual grau de relevância para a comunidade científica, domínio de pesquisa antigo ou recente
País	Comunidades científicas mais interessadas no tema, centros de pesquisa mais especializados no domínio, grupos de pesquisa com maior valor colaborativo
Autores	Pesquisadores de maior reconhecimento na área, que identifica as maiores autoridades no domínio
Veículo de publicação	Identificar os <i>journals</i> , congressos, <i>workshops</i> e afins de maior relevância para a comunidade interessada
Tipo de contribuição	Abordagens tomadas como tendências, identificar o estado-da-arte
Ambiente de ensino	Avaliar o uso de tecnologias para criar LMSs móveis ou ubíquos
Tecnologias IoT	Tendências de IoT para o domínio, tecnologias com soluções de utilização mais disponíveis

Fonte: Dados da pesquisa.

3.1.3 Bases de dados e critérios de inclusão/exclusão

Algumas bases de dados foram selecionadas para realizar a consulta, utilizando a *string* genérica apresentada. Para cada base de dados, a *string* foi adaptada de acordo com a sintaxe exigida pelo seu motor de busca, no entanto sem alterar a estrutura da *string* genérica. A Tabela 3 apresenta o volume de artigos recuperados, em primeiro momento, após a consulta a cada base de dados.

Para auxiliar na triagem dos estudos, foram definidos critérios de inclusão e exclusão. Os critérios de inclusão definidos reforçam a aceitação de um estudo sob análise, enquanto os critérios de exclusão definem por quais motivos um estudo não pode ser incluído no corpo

Tabela 3 – Resultados da primeira consulta às bases de dados

Base de dados	Estudos recuperados
ACM Digital Library	21
IEEE Xplore	253
ScienceDirect	590
Scopus	758
SpringerLink	2729

Fonte: Dados da pesquisa.

de estudos primários selecionados. Para os critérios de inclusão, apenas um foi definido para reforçar a aceitação dos estudos que trabalham com sistemas *U-learning*. Isso se deve ao grau de mobilidade e inteligência requisitada da infra-estrutura tecnológica do sistema, que aponta fortemente para o uso de tecnologias IoT.

Para os critérios de exclusão, foram definidos os seguintes:

- Todos os estudos que não tratem de IoT ou *E-learning* serão removidos.
- Caso haja um conjunto de estudos próximos de iguais, apenas um destes será incluído, excluindo os demais. Este critério se estende aos estudos duplicados.
- Todos os resultados que sejam sumários de livros serão excluídos. Estes serão reservados para uma avaliação separada, para determinar futuras tendências de domínio.
- Todos os estudos que não puderem ser lidos, por indisponibilidade ou estarem escritos em língua não compreendida pelos revisores, serão excluídos.

3.2 Condução

Nesta fase, foi realizada a leitura dos estudos recuperados das bases de dados consultadas. Como o volume de estudos a serem lidos era elevado, a fase de condução foi dividida em quatro etapas, em que os estudos foram avaliados segundo os objetivos definidos e os critérios de inclusão e exclusão. A divisão da fase de condução em etapas visa a identificação dos estudos a serem excluídos demandando um mínimo volume de leitura necessário para tal, tornando todo o Mapeamento Sistemático mais rápido.

3.2.1 Primeira etapa

Nesta etapa, todos os estudos recuperados foram avaliados de acordo com o título e resumo. Algumas das bases de dados consultadas (ex. SpringerLink) recuperam grande volume

de estudos sem nenhuma relação com o tema de pesquisa desejado. Desta forma, a leitura do título e resumo é suficiente para identificar tais estudos, e excluí-los prontamente. Ao longo desta etapa, também são removidos os resultados duplicados ou muito próximos.

3.2.2 Segunda etapa

Nesta etapa, os estudos foram lidos de forma superficial. Título e resumo foram lidos novamente, no entanto com o acréscimo dos capítulos de introdução e conclusão. Por fim, também foram consultados os títulos de seções e subseções, assim como legendas de figuras, tabelas e afins. Esta leitura mais atenta permite identificar e excluir quaisquer artigos não relacionados ao tema deste SM que tenham passado pela primeira etapa de leitura.

Ao final desta etapa, os estudos restantes estão relacionados com o mapeamento feito, restando avaliar se tal relação aparece apenas superficialmente ou se mantém durante o entendimento mais completo do trabalho descrito.

3.2.3 Terceira etapa

Nesta etapa, os estudos restantes foram lidos por completo. Esta leitura identifica os estudos a serem definitivamente selecionados, bem como os estudos que se relacionam com o tema do SM mas não atendem a todos os critérios de inclusão e exclusão. Ao final desta última etapa, os estudos restantes são classificados como estudos primários do mapeamento, e suas informações são utilizadas para sintetizar os resultados desejados.

3.3 Sumarização dos Dados

Durante a avaliação dos estudos, uma tabela com os resultados parciais foi mantida atualizada, a fim de manter controle sobre o progresso realizado. A Tabela 4 apresenta o número de estudos selecionados durante o MS. Nesta seção, é feita a discussão dos dados coletados e avaliados dos 124 estudos selecionados.

3.3.1 Ano

Ao avaliar os estudos selecionados de acordo com o ano de publicação, foram identificados volumes significativos nos anos de 2008, 2010 e 2011. Em contrapartida, houve um decréscimo significativo no ano de 2009. Estes números identificam que o problema de pesquisa investigado é recente, no entanto não muito. Como parte significativa dos estudos foi publicada anteriormente a 2010, é possível compreender o motivo pelo qual o termo IoT é raramente citado.

Apesar do decréscimo significativo no ano de 2009, é notável o aumento no volume de estudos publicados. Pode-se fundamentar este aumento pela popularização das tecnologias

Tabela 4 – Estudos selecionados por etapa de leitura.

Base de dados	Primeira etapa	Segunda etapa	Terceira etapa
ACM Digital Library	18	14	13
IEEE Xplore	147	72	52
ScienceDirect	112	36	28
Scopus	55	21	17
SpringerLink	243	27	14
TOTAL	575	170	124

Fonte: Dados da pesquisa.

móveis e pelo avanço das pesquisas em tecnologias de computação ubíqua. A [Figura 5](#) mostra estas informações.

Figura 5 – Ano de publicação dos estudos selecionados.

3.3.2 País

Ao avaliar os estudos segundo os países de origem, pode-se destacar prontamente Taiwan como o país onde mais se publicou estudos no tema de pesquisa investigado, somando, sozinho, 28,22% dos estudos selecionados. Seguido de Taiwan, também é possível notar o Japão como o

segundo país a publicar mais estudos sobre o tema, somando 12,09% do total. Ao somar com o volume de estudos publicados na China e na Coreia do Sul, identifica-se o oriente asiático como região que mais pesquisa aplicações de IoT para a educação, somando, apenas com os quatro países citados, 54,16% dos estudos selecionados.

Apesar da dominância da comunidade científica asiática, também foi identificado grande interesse da comunidade científica europeia. Como países com maior destaque na comunidade europeia, estão a Espanha (7,5%), Alemanha (5,64%) e Itália (5%). Com alguns outros países com contribuição marginal para o volume de estudos, a Europa fica em segundo lugar como região onde o tema de pesquisa é valorizado pela comunidade científica.

Por fim, é possível identificar os Estados Unidos com alguns estudos publicados (4,8%), o Brasil (2,4%) e alguns estudos de países da África. Os dados analisados podem ser verificados na [Figura 6](#).

Figura 6 – País de origem dos estudos selecionados.

3.3.3 Ambientes de aprendizagem

Quanto aos tipos de ambientes de aprendizagem desenvolvidos nos estudos selecionados, tem-se um volume claramente maior de ambientes U-learning, que somam 60,5% dos estudos. Dos ambientes restantes, vê-se 27,4% dos estudos tratando de ambientes M-learning, 9,7% tratando de ambientes E-learning, e os estudos restantes não tratando do desenvolvimento de um ambiente de aprendizado. Estes dados aparecem na [Figura 7](#).

É compreensível obter estes números se for considerado o mesmo fator apresentado na análise por ano de publicação: com o volume de estudos publicados crescendo nos últimos anos, e considerando que este tema de pesquisa tornou-se popular apenas em 2008, é importante

lembrar a presença das tecnologias móveis e ubíquas durante o desenvolvimento dos trabalhos apresentados. Desta forma, a falta de popularidade do tema antes da disseminação das tecnologias móveis e ubíquas fez com que o volume de ambientes E-learning desenvolvidos fosse realmente menor. Além disso, também é possível relacionar o crescimento no volume de estudos com as possibilidades criadas pelas novas tecnologias, que permitiram desenvolver ambientes M-learning e U-learning de forma diferente.

Figura 7 – Ambientes de aprendizagem dos estudos selecionados.

3.3.4 Tipos de contribuição

Quanto ao tipo de contribuição apresentada, a maioria significativa dos estudos apresenta uma ferramenta e/ou sistema como contribuição, somando cerca de 56,45% dos estudos selecionados. Em segundo lugar, encontram-se os *surveys* (12,09%), e em terceiro os *frameworks* (8,87%). Mesmo com menor expressão, também é possível notar a significância das arquiteturas e métodos. Estes dados podem ser vistos na [Figura 8](#).

Com o objetivo do MS em identificar as formas com que IoT é aplicada na educação, é compreensível o número de ferramentas encontradas, pois era esperado que fossem aceitos vários estudos com aplicação dos conceitos de IoT na prática. Quanto aos *surveys*, também é compreensível uma vez que apresentam uma revisão de artigos relevantes ao tema de pesquisa, e, assim, apresentam formas de aplicação de IoT. O aparecimento de *frameworks* nesta lista pode ser explicado por que parte dos trabalhos mostram a preocupação em propor *frameworks*, aplicações “semi-completas” e reutilizáveis que podem ser especializadas em aplicações customizadas (FAYAD; SCHMIDT, 1997), que facilitem o desenvolvimento de plataformas de aprendizado. Mesmo em menor expressão, a presença de arquiteturas e métodos demonstra a preocupação da comunidade científica em explorar aspectos tanto teóricos quanto práticos do desenvolvimento deste tipo de sistema.

Figura 8 – Tipos de contribuição dos estudos selecionados.

3.3.5 Recursos educacionais

É significativo o número de estudos que se utilizam de textos como recurso educacional, somando 37,09% do total. Os ambientes de aprendizado vêm em seguida, somando 30,64%. O uso de imagens soma 24,19% do total. 33,87% referem-se aos estudos onde nenhum recurso educacional é utilizado, e os recursos restantes aparecem em número expressivamente menor. Conforme é evidenciado pela soma das porcentagens apresentadas, que excede 100%, poucos estudos utilizam apenas um recurso educacional. Desta forma, a utilização de mais de um destes recursos em um mesmo estudo é frequente. Os dados citados são apresentados na [Figura 9](#).

O grande volume de estudos que não utilizam um recurso educacional ocorre por dois fatores distintos: a presença de vários estudos de teor puramente teórico, e a proposta de sistemas que permaneceram apenas em propostas. Como o foco do MS era na aplicação das tecnologias IoT, era esperado que alguns estudos não se utilizassem de algum recurso educacional.

3.3.6 Tecnologias IoT

Conforme apresentado na [Figura 10](#), a presença de dispositivos *Portable Digital Assistant* (PDA) é significativa. Um PDA é um dispositivo portátil de mão, em geral um *smartphone*. Estes somam 63,7% de presença nos estudos, seguidos das WSNs (48,38%) e das etiquetas RFID (42,74%). A utilização da nuvem como tecnologia fundamental aparece em apenas 12,09% dos casos. É notável, também, uma pequena fração onde nenhum dispositivo IoT é utilizado, e uma presença marginal dos Códigos *Quick Response* (QR Codes)¹.

¹ É uma espécie de código de barras, porém com 2 dimensões. Geralmente, pode ser escaneado por um *smartphone* e convertido em um texto, URI, número de telefone, e-mail ou contato, não se limitando a estas opções.

Figura 9 – Recursos educacionais utilizados nos estudos selecionados.

É compreensível a presença de PDAs, WSNs e etiquetas RFID em fatias maiores dos estudos. WSNs e RFID são tecnologias que foram popularizadas com as primeiras pesquisas em IoT, assim sendo referências de tal tipo de tecnologia. Já os PDAs aparecem pela popularização do uso dos mesmos, no entanto através da evolução dos ambientes de aprendizagem, principalmente pelos ambientes móveis. A presença dos QR Codes pode ser entendida uma vez que o uso de PDAs é comum, e que estes dispositivos contam cada vez mais com câmeras fotográficas menores e melhores.

3.3.7 Utilização de serviços

Com a proposta de trabalho envolvendo o uso de LPS e serviços, é importante destacar quaisquer estudos encontrados que apresentem a utilização destas abordagens. Isto permite identificar prontamente desafios já encontrados no desenvolvimento de sistemas de aprendizagem, bem como as contribuições e limitações já definidas. Ao todo, 3 trabalhos foram encontrados envolvendo o uso de serviços, enquanto nenhum apresentou o uso de LPS. Estes trabalhos são apresentados a seguir.

O trabalho de [Badr e Maamar \(2009\)](#) apresenta uma SOA, projetada para sistemas *E-learning*. É importante notar que a SOA proposta é baseada no conceito de *grid computing*, assim tratando-se de uma arquitetura para sistemas distribuídos. Apesar de não apresentar uma forma de utilizar dispositivos IoT, a SOA proposta contém todas as características necessárias para oferecer suporte a este tipo de dispositivo, bem como características operacionais capazes

Figura 10 – Tecnologias IoT utilizadas nos estudos selecionados.

de oferecer funcionalidades concordantes com sistemas de apoio à IoT.

Já o trabalho de [Kim, Song e Yoon \(2011\)](#) considera o desenvolvimento de sistemas de aprendizado utilizando serviços, porém não se aprofunda nesta teoria. Neste estudo, é considerado o desenvolvimento de um sistema orientado a serviços, uma vez que o objetivo dos autores é investigar o desenvolvimento de sistemas de aprendizado inteligentes baseados em serviços hospedados na *nuvem*. Os autores definem a arquitetura para o modelo computacional proposto, bem como componentes inteligentes baseados no conceito de elasticidade.

Por fim, o trabalho de [Villanueva et al. \(2015\)](#) apresenta uma arquitetura para sistemas de aprendizado cientes de contexto, utilizando serviços. A arquitetura é uma SOA, e os dispositivos IoT utilizados são classificados pelos autores segundo seu nível de inteligência e capacidade computacional. Esta arquitetura é apresentada segundo os padrões *web* em que os serviços são utilizados e os dispositivos IoT que fornecem a ciência de contexto, bem como de que forma eles são inseridos na SOA.

3.4 Características dos Estudos Selecionados

Como o SM descrito neste capítulo apresentou um grande volume de estudos selecionados, decidiu-se por filtrar os estudos a serem apresentados neste documento pelo nível de relação com o trabalho apresentado no [Apêndice B](#). Para realizar esta seleção, foram utilizados os seguintes critérios:

- **Critério de Filtragem 1:** Os estudos onde o paradigma da Internet das Coisas é mencionado serão incluídos. Esta decisão se baseia no uso importante de dispositivos IoT para o

trabalho proposto, visando uma melhor coleta de informações para a construção dos perfis de contexto dos usuários.

- **Critério de Filtragem 2:** Os estudos onde *smartphones* são utilizados como dispositivos IoT serão incluídos. Não atendem a este critério os estudos onde os *smartphones* são utilizados apenas como interfaces de acesso portáteis.

Os 41 estudos seleccionados por estes critérios são listados após a [Seção 3.5](#), no final deste capítulo. A seguir, alguns são resumidos, após serem agrupados e divididos segundo características encontradas. As informações de todos os estudos seleccionados neste SM são apresentados em uma planilha, no [Apêndice A](#).

3.4.1 Middlewares

Dois dos estudos filtrados apresentam um *middleware* como contribuição. Este tipo de sistema funciona como um intermediador entre sistema operacional e aplicações de usuário, geralmente fornecendo suporte ao desenvolvimento e operação de sistemas para um nicho específico.

O trabalho de [Boyinbode e Bagula \(2011\)](#) apresenta um *middleware* de suporte ao desenvolvimento de sistemas de aprendizagem ubíquos, com foco no suporte para pessoas com deficiências. Os autores afirmam que o *middleware* possui um modelo capaz de armazenar informações sobre a deficiência de um usuário, e os tipos de mídias mais adequadas a ele.

Já o trabalho de [Fiore et al. \(2013\)](#) tem foco no suporte à criação de sistemas de aprendizagem, criando um *middleware* que ofereça mais ferramentas aos usuários. Segundo os autores, este *middleware* permite a criação de cenários de aprendizado utilizando *mixed reality*, isto é, mecanismos de interação entre ambientes reais e virtuais, integrando ambientes de aprendizado virtuais com experiências em ambientes reais. A criação de cenários de aprendizado deste *middleware* é feita através da criação de objetos de aprendizado, e da vinculação destes a sensores integrantes de uma WSN.

3.4.2 Idiomas

Dos estudos filtrados, cinco tratam de sistemas de apoio ao aprendizado de idiomas. O trabalho de [Paletta et al. \(2013\)](#) utiliza jogos, auxiliando imigrantes no aprendizado do idioma do país onde moram. Os autores afirmam que um dos objetivos deste jogo, nomeado MASELTOV, é auxiliar tais imigrantes não apenas a aprender o idioma do país onde irão morar, mas também aspectos culturais locais. Isto é alcançado pela interação com outros jogadores, e mecanismos de ciência de contexto.

O trabalho de [Ogata \(2008\)](#) apresenta uma série de sistemas de aprendizado em que o autor esteve envolvido, lidando de diversas formas com a aplicação de tecnologias IoT no

aprendizado de idiomas e aspectos culturais. No entanto, neste trabalho é apresentada a proposta de um novo sistema, o LORAMS, que utiliza fundamentalmente de etiquetas RFID e PDAs para criar cenários de aprendizado, permitindo que os usuários gravem vídeos e os vinculem a alguma etiqueta, criando, assim, um ambiente de aprendizado colaborativo. O mesmo autor participou de outro projeto, apresentado no trabalho de Hou *et al.* (2010). No entanto, neste trabalho é apresentada uma evolução do JAMIOLAS, sistema voltado ao aprendizado de gestos e onomatopeias típicos da cultura japonesa.

O trabalho de Chen *et al.* (2009) apresenta um sistema simples de aprendizado: os usuários aprendem o idioma através da escrita de ensaios. No entanto, o sistema oferece conteúdos relacionados ao tema do ensaio, através do contexto do usuário medido por meio de etiquetas RFID e PDAs. Assim, o usuário tem experiências situadas de aprendizado. É importante destacar que os autores avaliaram este sistema com alunos do ensino fundamental, para o aprendizado de sua língua materna.

O trabalho de Wu *et al.* (2014) apresenta um sistema de assistência à leitura de textos, armazenando, também, as atitudes de cada usuário diante dos artefatos recomendados. Neste sistema, o local de um usuário é detectado via etiqueta RFID, e cada situação de aprendizado é armazenada no portfólio do usuário.

3.4.3 *Computação e Mecatrônica*

Um dos estudos filtrados apresenta uma plataforma de ensino, que foi avaliada em cursos de mecatrônica. O trabalho de Chao *et al.* (2015) apresenta um sistema, nomeado CLEM, que oferece infra-estrutura lógica para a montagem de uma plataforma de ensino que ofereça suporte, principalmente, à integração dos usuários através do uso de ambientes similares a fóruns, onde dúvidas podem ser postadas por alunos e solucionadas por docentes e especialistas. O sistema também facilita a comunicação entre participantes, incentivando o aprendizado colaborativo. É importante ressaltar que o CLEM é um sistema criado nos moldes de um *Software as a Service* (SaaS), isto é, os usuários acessam todo o sistema através da internet, onde o sistema fica disponível, utilizando o paradigma da “computação na nuvem”.

O trabalho de Villanueva *et al.* (2015) também aparece nesta categoria, sendo avaliado em cursos de eletrônica digital. O resumo deste trabalho é apresentado na Subseção 3.3.7.

3.4.4 *Smart Campus*

Smart campus é uma forma de melhorar a qualidade de vida dos estudantes que frequentam uma universidade, no que se refere ao desempenho acadêmico. O *campus* é equipado com dispositivos inteligentes, a exemplo dos IoT, que são capazes de coletar informações sobre cada ambiente do *campus*. Desta forma, a administração da universidade pode tornar os serviços administrativos mais inteligentes, assim como novos serviços podem ser criados. Em relação ao

aprendizado dos alunos, novas formas podem ser implementadas em experiências práticas.

O trabalho de [Atif e Mathew \(2013\)](#) apresenta uma especificação de *framework* para *smart campus*, destacando os processos e locais de aprendizado, o modelo de aprendizagem ubíqua, os metadados que devem ser considerados e as características da colaboração entre usuários. Os autores não apresentam um experimento onde este *framework* é avaliado, no entanto compensando com riqueza nos detalhes dos modelos e processos.

Já o trabalho de [Alghamdi e Shetty \(2016\)](#) mostra os resultados de um *survey* feito sobre este tópico. Os autores apresentam algumas das principais formas pelas quais *smart campi* são implementados. Entre elas, pode-se mencionar prédios inteligentes, malhas inteligentes, ambientes de aprendizado e gerenciamento de água e esgoto. Os autores também discutem quais os desafios em construir um *smart campus*, bem como as oportunidades envolvidas.

Por fim, o trabalho de [Coccoli et al. \(2014\)](#) discute o grau em que as universidades ao redor do mundo implementam os *smart campi*. Segundo os autores, a maioria dos *campi* de universidades já são considerados *smart campi*, uma vez que utilizam tecnologias contemporâneas integradas a sistemas de informação. No entanto, é possível evoluir um *smart campus* para um *smarter campus*, através da implementação de funcionalidades que apoiem a administração do *campus*, como ferramentas para análise dos dados e módulos de apoio à tomada de decisão. A contribuição dos autores está no modelo de processo definido, que auxilia a análise das necessidades das pessoas envolvidas e dos objetivos a serem alcançados.

3.4.5 Visitação de Museus

Alguns dos estudos filtrados apresentam, indiretamente, formas de implementar soluções que facilitem a visitação a museus. O trabalho de [Ogata e Yano \(2004\)](#) é um bom exemplo disso, quando apresenta o TANGO, sistema que identifica etiquetas RFID próximas ao usuário e apresenta os conteúdos educacionais correspondentes. Ainda segundo os autores, para avaliar este sistema, foi feito um experimento em que este era necessário para localizar objetos distribuídos em um ambiente fechado. Extrapolando tais objetos para itens de um museu, fica evidente de que o TANGO é utilizável em um ambiente deste tipo para auxiliar os visitantes a identificar os itens em exibição.

O trabalho de [El-Bishouty, Ogata e Yano \(2008\)](#) apresenta características semelhantes, referentes ao sistema apresentado, o PERKAM. No entanto, este considera a conexão à internet como fator importante para o funcionamento da aplicação. Além disso, os autores também mencionam as limitações dos PDAs utilizados como cruciais para determinar as otimizações necessárias ao sistema. Este estudo contribui valiosamente para o domínio dos sistemas de auxílio à visitação de museus pela preocupação com a otimização do sistema, levantando alguns requisitos não funcionais relacionados como críticos para sistemas que usam PDAs.

O trabalho de [Hwang et al. \(2011\)](#) apresenta um uso mais interativo para este tipo de

sistema. Utilizando a identificação de características em borboletas como tópico de aprendizado, os autores apresentam um sistema baseado em PDAs, em que os usuários devem navegar por um dado ambiente à busca de borboletas de espécies específicas. O sistema foi experimentado com estudantes do ensino primário, com o objetivo de investigar a carga de aprendizado adquirido através do sistema e o grau de compartilhamento entre os sujeitos. Os resultados indicam que o uso do sistema proposto potencializa as atividades educacionais realizadas através do mesmo, aumentando a carga teórica aprendida pelos usuários. No caso deste estudo, o sistema não apenas auxilia os usuários a identificar os objetos de estudo, como automatiza processos de avaliação para mensurar o conhecimento adquirido. Assim, os usuários são informados, logo após à avaliação, do resultado obtido.

3.5 Considerações Finais

O SM realizado permitiu identificar 124 estudos onde tecnologias IoT são aplicadas em sistemas de aprendizagem. Apesar de alguns estudos utilizarem serviços, nenhum estudo selecionado apresentou o uso de linhas de produtos. O uso de etiquetas RFID, WSNs e *smartphones* é frequente, o que mostra maior facilidade em desenvolver aplicações cientes de contexto com estas tecnologias. Mais da metade dos sistemas encontrados referem-se a ambientes *U-learning*, contendo um grande volume de exemplos de sistemas onde a ciência ao contexto é implementada. Desta forma, há vários exemplos onde tecnologias IoT são utilizadas com todo o seu potencial.

Com os resultados coletados neste SM, fica evidente de que a proposta de uma LPOS para sistemas de aprendizagem cientes de contexto é plausível, sendo útil para o desenvolvimento de uma linha de produtos. Além disso, segundo o volume de resultados que mencionam os *smart campus* e sistemas de auxílio à visita de museus, o desenvolvimento de uma LPOS para este domínio específico de aplicações aparenta ser consideravelmente atrativo para a comunidade científica.

Também é importante notar o trabalho de Junior (2015), que não é mapeado por este SM segundo as *strings* de pesquisa, porém é recomendado por especialistas do domínio. Conforme apresentado no Capítulo 1, este trabalho introduz uma LPS desenvolvida para instanciar sistemas *M-learning*. Ao considerar os dados deste estudo junto aos resultados apresentados neste capítulo, é evidenciada ainda mais a importância de conduzir pesquisas acerca de tal categoria de sistemas de aprendizagem.

Por fim, como atualmente a tecnologia dos *smartphones* tornou-se popular, e estes podem ser considerados dispositivos IoT completos, fica delineada a necessidade de explorar o uso de *smartphones* como dispositivos IoT para sistemas de aprendizagem, em especial os de auxílio à visita de museus. Tal necessidade é explorada no capítulo seguinte, onde uma proposta para melhorar o reuso de *software* neste domínio de aplicações é apresentada.

A LINHA DE PRODUTOS: MTTPL

Segundo os resultados encontrados no SM, descrito no [Capítulo 3](#), existem diversos estudos que apresentam propostas de modelagens diferentes para sistemas *e-learning* com o uso de dispositivos IoT ([HSU; LAI; YU, 2008](#); [CABALLÉ, 2015](#); [LIU et al., 2007](#); [CHANG et al., 2015](#)). No entanto, poucos estudos dentre os selecionados baseiam-se em modelagens propostas previamente. Desta forma, é evidenciada a carência de uma modelagem que possa ser usada como referência, a fim de amenizar o esforço necessário para modelar novos sistemas deste domínio de aplicação.

Ainda segundo as características dos sistemas encontrados pelo SM, é necessário considerar a presença de dispositivos IoT em tal modelagem, já que possuem função fundamental para que as soluções encontradas alcancem seus objetivos, conforme explicado no [Capítulo 2](#). Assim, para resolver o problema supracitado, é necessário aplicar uma técnica de reúso de *software* que permita a incorporação de dispositivos IoT, além de possibilitar uma fácil (re)configuração desses dispositivos perante o sistema. Tomando estas características como motivação, foi escolhida a técnica de LPOS ([LEE; KOTONYA, 2010](#)).

A abordagem de LPOS é uma especialização da técnica de LPS ([GOMAA, 2005](#)), incorporando os conceitos de orientação a serviços ([ROSEN et al., 2008](#)). Com o mapeamento de pontos de variabilidade e instanciação condicionada de *assets* da LPS, e a utilização de sistemas RESTful e independência operacional dos serviços, é possível elaborar instâncias de linhas de produtos de *software* capazes de comportar subsistemas independentes, além de bem conhecidos e descritos. Além disso, com a possibilidade de usar protocolos HTTP para a comunicação entre os serviços, a integração entre estes serviços é mais simples. Também é simples integrá-los com dispositivos IoT já que eles podem se comunicar com os serviços pelas mesmas APIs previamente conhecidas e mapeadas.

Por fim, o escopo ainda é amplo demais segundo o domínio de aplicações. Como o SM sumariza soluções encontradas para diferentes domínios de aplicação, mapear tal variedade

de domínios para a modelagem de uma única aplicação é excessivamente custoso, complexo e propenso a falhas. Portanto, para eliminar tais excessos, é necessário escolher um domínio de aplicação abrangido pelo SM. No caso deste trabalho, optou-se pelos sistemas de apoio à navegação de museus, que focalizam o problema da localização dos usuários dentro do museu a ser visitado, oferecendo ferramentas que os auxiliem a entender sua posição atual. A visita a um museu pode ser considerada como uma atividade de aprendizado, conforme mencionado no [Capítulo 1](#).

Neste capítulo, é apresentada a *Museum Touring and Tutoring Product Line* (MTTPL): uma proposta para solucionar os problemas de retrabalho identificados após analisar os dados do SM. Esta proposta se baseia no uso da técnica de LPOS para melhorar o grau de reúso de *software*. Além disso, conforme os problemas citados acerca da escolha do domínio de aplicação, a MTTPL tem enfoque nos sistemas de auxílio à visita de museus, sendo proposta como um meio de melhorar o reúso de *software* para este domínio específico de aplicações.

Este capítulo é organizado da seguinte maneira: na [Seção 4.1](#), é dada uma rápida definição do que é a MTTPL e quais seus objetivos gerais. Na [Seção 4.2](#), é apresentado o processo seguido para o desenvolvimento da linha de produtos. Na [Seção 4.3](#), é feita a relação dos resultados do SM com os requisitos utilizados como base para a MTTPL. Na [Seção 4.4](#), são apresentados os casos de uso e um modelo ilustrando-os. Na [Seção 4.5](#), são apresentadas as entidades envolvidas na modelagem da MTTPL, apresentando-as na forma de um modelo conceitual. Na [Seção 4.6](#), são apresentadas as características da linha de produtos de *software* e sua organização em um modelo de *features*. Na [Seção 4.7](#), são apresentados os serviços derivados das características da MTTPL, com uma descrição dos papéis de cada um e suas relações, culminando num modelo que ilustra a organização dos serviços gerados. Na [Seção 4.8](#), são apresentados os *assets* gerados pela MTTPL, mapeando os serviços e *features* para classes e campos a serem implementados. Por fim, na [Seção 4.9](#) são feitas as considerações finais deste capítulo.

4.1 Definição

A MTTPL é uma LPOS, idealizada a partir dos resultados encontrados no SM apresentado no [Capítulo 3](#). Esta linha de produtos de *software* visa, primordialmente, a redução no esforço em instanciar novos produtos de *software* para auxiliar a visita de museus, utilizando dispositivos IoT.

Para cumprir seu papel, a MTTPL deve obedecer aos seguintes princípios:

- Permitir a instanciação de sistemas que auxiliem a visita de museus;
- Facilitar a incorporação de dispositivos IoT à modelagem proposta;

Devido à presença de dispositivos IoT, os produtos instanciados a partir da MTTPL devem ser cientes do contexto dos usuários.

4.2 Processo Adotado

Conforme dito anteriormente, o PLUS (GOMAA, 2005) é um método de desenvolvimento de LPS, que utiliza a UML como linguagem de representação. Devido ao uso desta linguagem, o PLUS estabelece que a linha de produtos resultante seja orientada a objetos.

No entanto, para o contexto da abordagem de LPOS, o PLUS não é utilizável com todos os seus benefícios, dado que não é considerada a modelagem do sistema como um agregado de serviços. Desta forma, se faz necessário adaptar o PLUS para que seja possível modelar cada um dos serviços da MTTPL de forma independente, sem perder o poder de representação que o PLUS traz ao projeto.

A abordagem adotada para solucionar este problema foi utilizar o método proposto por Lee, Muthig e Naab (2008), apresentado no Capítulo 2. A proposta baseia-se no agrupamento das características da LPS, visando delinear o escopo dos diferentes serviços que podem instanciar as características de tais grupos. A abordagem é baseada na definição de componentes arquiteturais da linha de produtos, mas que, por definição, podem, também, ser pensados como serviços.

4.3 Requisitos

O primeiro passo na modelagem da LPOS é levantar os requisitos desejados para a mesma. Para isto, utilizou-se os estudos selecionados no SM como fonte de requisitos.

Apesar do SM possuir um escopo mais abrangente, os sistemas de visitação a museus figuram como parte integrante dos estudos do SM. Desta forma, é seguro utilizar tais estudos para extrair os requisitos desejados para a MTTPL. Assim, foi possível utilizar os resultados do SM como ponto de partida para a modelagem da MTTPL.

Os dados extraídos, utilizados como requisitos para a MTTPL, foram compilados na tabela apresentada no Apêndice A, que apresenta as características extraídas acerca dos resultados apresentados nos trabalhos selecionados. Além destes dados, também foi utilizada a análise feita por Junior (2015).

Por fim, foram coletados requisitos do Museu da Fauna e da Flora do ICMC. Como este trabalho foi realizado dentro do Instituto onde este museu é localizado, bastou entrar em contato com os membros curadores do mesmo, e analisar como o mesmo é organizado e mantido. Também foi avaliado como a visita ao museu foi idealizada pelos curadores, a fim de mapear tais características nos requisitos da MTTPL e levantar novas *features*.

Após a coleta de todos os requisitos a serem utilizados para elaborar a MTTPL, foi

realizada uma missão para refinar tais requisitos e gerar um conjunto validado por especialistas do domínio. Para isto, foi escolhido o Núcleo de Excelência em Tecnologias Sociais (NEES) da Universidade Federal de Alagoas, o qual coopera com o ICMC-USP no contexto de um projeto Procad-CAPES¹. O autor desta dissertação realizou uma visita ao NEES, em que foi realizada a validação e aprimoramento dos requisitos do sistema.

Dentre os requisitos, foram identificados os seguintes como comuns entre todos os estudos selecionados, configurando-se, portanto, como mandatórios:

- Presença de um mecanismo de assistência à visita;
- Utilização de algum tipo de mídia de conteúdo (e.g. texto, vídeo, som);
- Utilização de algum dispositivo para sensoriamento do contexto dos visitantes, relativos ao ambiente.

Assim, foram encontrados os seguintes requisitos classificados como opcionais:

- Automação na geração das rotas de visita;
- Sugerir rotas aos visitantes;
- Aplicação de avaliações sobre o conhecimento adquirido durante uma visita;
- Exibição de detalhes da rota até um dado item do museu;
- Ferramentas amigáveis para autoria de conteúdos;
- Ferramentas para visitantes criarem conteúdos;
- Mídias de conteúdo: texto, áudio, vídeo, imagem;
- Sistemas de recomendação e sugestão de itens para visitar;
- Traçagem dos perfis dos visitantes, incluindo informações de interesses e aprendizado (prévio e adquirido);
- Rastreamento dos visitantes via RFID, Beacons ou GPS;
- Utilização de RFIDs de ambiente para detectar transição de visitantes entre ambientes;
- Utilização de sensores sem-fio: Arduino e Raspberry Pi;
- Utilização de *smartphones* com algum sensor interno: giroscópio, acelerômetro e termômetro.

¹ Projeto Procad-CAPES número 71/2013. Título: Pesquisa, Integração e Capacitação de Recursos Humanos em Tecnologias Educacionais e Engenharia de Software, Coordenador: Prof. Dr. José Carlos Maldonado

4.4 Diagrama de Casos de Uso

O próximo passo para a elaboração da MTTPL, segundo a abordagem escolhida, prevê a construção do diagrama de casos de uso. Para tal, as características encontradas no SM foram agrupadas, visando formar os diferentes casos de uso que as incorporem num produto final. O resultado pode ser visto na [Figura 11](#).

Figura 11 – Diagrama de casos de uso da MTTPL.

4.5 Modelo Conceitual

Visando melhorar o entendimento do domínio de aplicação da linha de produtos, optou-se por elaborar um modelo conceitual. Este modelo mapeia entidades e conceitos do sistema, bem como suas relações. Desta forma, é possível compreender melhor o papel de cada elemento presente na aplicação.

Para realizar este mapeamento, foi utilizado o método apresentado por [Larman \(2002\)](#): tais entidades e conceitos foram identificados a partir da descrição dos casos de uso do sistema, bem como as relações entre os mesmos. O resultado desta análise é apresentado na [Figura 12](#).

Figura 12 – Modelo conceitual da MTTPL.

powered by Astah

4.6 Features

Para finalizar a modelagem da MTTPL, era necessário elaborar o modelo de características. Utilizando os dados da Seção 3.5, bem como os modelos das Figuras 11 e 12, foi possível elaborar a seguinte lista de características:

- **Ciência de Contexto:** grupo de *features* que trata dos mecanismos responsáveis por agregar ciência de contexto ao produto final;
- **Sistema de Recomendação:** característica responsável por sinalizar a presença de um sistema de recomendação no produto final;
- **Informação de Contexto:** *feature* responsável por agregar todas as informações coletáveis pelo sistema, e que podem ser usadas para determinar o contexto dos usuários;
- **Perfil de Usuário:** grupo de *features* que agrega as informações determinantes de contexto exclusivas dos usuários;
- **Localização:** *feature* que define a presença da (geo)localização dos usuários como informação de contexto;
- **RFID de Local:** *feature* que indica a presença de dispositivos RFID como sensores de localização para os usuários;

- **GPS:** característica que indica o uso de sensores GPS como localizadores;
- **Beacon:** *feature* que indica a presença de *beacons* como sensores de localização;
- **Aprendizado Prévio:** característica que indica a utilização de informações que os usuários tinham previamente ao uso do sistema;
- **Aprendizado Adquirido:** semelhante ao *Aprendizado Prévio*, mas trata apenas do aprendizado adquirido através do uso do sistema;
- **Interesses:** *feature* que indica o uso de um conjunto de interesses dos usuários como ferramenta de contexto;
- **Ambiente de Inserção:** grupo de características que sinaliza o uso de informações sobre o ambiente de inserção do usuário como fonte de contexto, e que agrega as tecnologias usadas para tal;
- **Sensores de Ambiente:** grupo de características que engloba todas as tecnologias (previstas na modelagem da MTTPL) utilizáveis para sensoramento do ambiente onde os usuários estarão utilizando o sistema;
- **RFID de Ambiente:** *feature* que sinaliza o uso de sensores RFID para determinar o ambiente onde o usuário está;
- **Sensor Sem-fio:** grupo de características que indica a presença de dispositivos sensores sem-fio. Também é o grupo de características que pode incorporar novos dispositivos IoT não previstos em outros grupos de *features* mais especializados;
- **Raspberry Pi:** característica que sinaliza a presença de dispositivos Raspberry Pi como sensores IoT;
- **Arduino:** característica que sinaliza a presença de dispositivos Arduino como sensores IoT;
- **Sensor de Smartphone:** grupo de características que indica o uso de *smartphones* como dispositivos IoT, e que engloba os sensores internos específicos a serem usados;
- **Giroscópio:** característica que sinaliza o uso do giroscópio de *smartphones*;
- **Acelerômetro:** característica que sinaliza o uso do acelerômetro de *smartphones*;
- **Termômetro:** característica que sinaliza o uso do termômetro de *smartphones*;
- **NFC PDA:** *feature* que sinaliza o uso de dispositivos portáteis de uso pessoal (*smartphones* ou não), com tecnologia NFC. É necessária para a instanciação das características que façam uso da tecnologia RFID;

- **Guia do Museu:** grupo de características que trata da apresentação dos conteúdos do museu ao usuário, bem como assistência de navegação e Experiência de Usuário, ou *User Experience* (UX);
- **Autoria de Conteúdos:** *feature* que sinaliza a presença de ferramentas de auxílio para a autoria de novos conteúdos;
- **Autoria por Usuários:** *feature* que sinaliza a possibilidade dos usuários criarem novos materiais de aprendizado;
- **Mídias de Conteúdo:** grupo de características que engloba todas as mídias utilizáveis pelo sistema para entregar os conteúdos de aprendizado;
- **Imagem:** característica que indica a presença de imagens e fotos como material de aprendizado;
- **Texto:** característica que indica a presença de textos como material de aprendizado;
- **Áudio:** característica que indica a presença de sons e trechos de áudio como material de aprendizado;
- **Vídeo:** característica que indica a presença de vídeos como material de aprendizado;
- **Sugestão de Conteúdos:** *feature* que representa a presença de mecanismos de sugestão de conteúdos educacionais aos usuários. É dependente da *feature* *Sistemas de Recomendação*;
- **Visitação Assistida:** grupo de características que engloba todas as ferramentas oferecidas pelo sistema para auxiliar os usuários a realizarem uma visita ao museu de forma facilitada;
- **Rotas Automatizadas:** característica que sinaliza a automação das rotas a serem seguidas pelos usuários, por parte do sistema;
- **Sugestão de Rotas:** característica que sinaliza a recomendação de rotas entre itens do museu para os usuários;
- **Avaliação de Conhecimento:** característica que indica a presença de avaliações, para medir o conhecimento adquirido pelos usuários durante a visita;
- **Rota até Objeto:** característica que indica a presença de um indicador de localização dos itens a serem visitados pelos usuários;

A característica *NFC PDA*, apesar de mapeada na linha de produtos, é uma *feature lógica*: define comportamentos do produto final, sem causar alterações no código-fonte. Basicamente, é uma característica que só pode ser observada através do uso do sistema, avaliando a presença e/ou funcionamento de outras características que possam ser influenciadas por esta. No caso da *NFC PDA*, ela é necessária nos casos em que o produto instanciado faça uso da tecnologia de

RFID, segundo as *features* com tal sigla em seu nome. No contexto da MTTPL, ela é útil para indicar que os aparelhos usados pelos usuários devem ter a tecnologia NFC presente, para que outras características funcionem conforme o esperado.

Também é vital destacar a dependência entre as *features* *Sistema de Recomendação* e *Sugestão de Conteúdos*. Estas duas características diferem-se no impacto que refletem no produto instanciado: a *feature* *Sistema de Recomendação* indica apenas a presença de um sistema de recomendação, que deve ser usado na personalização dos conteúdos de aprendizado oferecidos aos usuários. Já a *feature* *Sugestão de Conteúdos* indica que o sistema irá recomendar conteúdos para os usuários visitarem, utilizando informações de contexto para tal, como, por exemplo, interesses e aprendizado adquirido. Desta forma, a MTTPL permite que um sistema de recomendação seja utilizado para personalizar os conteúdos educacionais apresentados de forma mais inteligente, sem necessariamente utilizar de tal para recomendar outros conteúdos.

Inicialmente, as características da MTTPL foram divididas em dois grupos distintos a partir da característica raiz, *MTTPL*, segundo um único critério: separar *features* relacionadas às atividades de ensino e aprendizagem das *features* relacionadas ao contexto dos usuários. Assim, foram definidos os grupos *Guia do Museu* e *Ciência de Contexto*, respectivamente. Outro agrupamento que é importante ser mencionado refere-se às informações coletadas sobre o contexto dos usuários, e como os sensores mapeados pela MTTPL seriam mapeados segundo tais dados.

Em um primeiro momento, as informações de contexto foram separadas em duas categorias, gerando os grupos de características *Perfil de Usuário* e *Ambiente de Inserção*. Como a posição do usuário relativa ao sistema é referente ao usuário e não ao ambiente, a *feature* *Localização* foi inserida em *Perfil de Usuário*. Já na característica *Ambiente de Inserção*, os sensores relativos aos dados do ambiente de operação do sistema foram divididos entre os distribuídos pervasivamente pelos ambientes, e aqueles embarcados nos *smartphones*, gerando as ramificações do grupo *Sensor de Ambiente*.

O resultado das *features* geradas a partir da identificação feita a partir dos modelos já gerados, bem como dos agrupamentos citados, foi compilado no modelo apresentado na [Figura 13](#). Este modelo apresenta visualmente as características listadas anteriormente, bem como suas relações. Assim, a visualização dos pontos de comunalidade e variabilidade da MTTPL é mais fácil, acelerando o entendimento destes.

4.7 Serviços

4.7.1 Definição dos Serviços

Para finalizar a modelagem da MTTPL, seria necessário agrupar as características do modelo de *features* em diferentes serviços. De acordo com a abordagem adotada ([LEE](#); [KO-](#)

Figura 13 – Modelo de *features* da MTTPL.

TONYA, 2010), para realizar esta seleção é necessário agrupar as *features* de acordo com os usos do sistema, de forma que os serviços resultantes reflitam funcionalidades geradas por tal agrupamento.

Desta forma, a partir do modelo de características da MTTPL, agrupando-as de acordo

com a abordagem adotada, foram derivados os seguintes serviços:

- **Provedor de Conteúdo:** serviço composto a partir do grupo de *features Mídias de Conteúdo*, tem a função de lidar com o armazenamento e provisionamento de mídias de conteúdo;
- **Mapeador de Rotas:** serviço composto pelo grupo de características *Visitação Assistida*, exceto a *feature Avaliação de Conhecimento*. Este serviço tem a incumbência de lidar com o traçado das rotas dos usuários, bem como prover as informações necessárias para o usuário se guiar pelo mapa do museu;
- **Provedor de Avaliações:** este serviço compreende as características *Avaliação de Conhecimento* e *Sistema de Recomendação*, e é responsável por prover avaliações de conhecimento ao sistema, bem como os respectivos resultados. Neste serviço também ficam concentrados os algoritmos de inteligência artificial mais avançados do sistema, referentes à implementação de uma ciência de contexto mais elaborada;
- **Recomendador de Itens:** serviço que incorpora apenas a característica *Sugestão de Conteúdos*, porém que fica responsável por prover sugestões de itens do museu a serem visitados pelo usuário;
- **Editor de Conteúdo:** serviço que engloba as características *Autoria de Conteúdos* e *Autoria por Usuários*, é responsável por prover ao sistema os CRUDs dos itens do museu, bem como gerenciar, quando disponível, quais informações cada usuário pode gerenciar. O principal objetivo deste serviço é permitir que o sistema forneça ferramentas de gerenciamento dos itens do museu, sem a necessidade de trabalhar os dados diretamente no banco de dados;
- **Broker de Sensores:** este serviço é o que intermedia a comunicação entre sensores IoT e o sistema. É composto pelo grupo de características *Sensores de Ambiente*, excluindo o subgrupo *Sensor de Smartphone*. Este serviço é responsável por coletar as informações dos sensores e alimentar o sistema com tais dados. É claro, ele também é responsável por realizar o pré-processamento dos dados, visando formatá-los num formato que os outros serviços possam reconhecer e interpretar;
- **Provedor de Usuários:** serviço que incorpora o grupo de características *Perfil de Usuário*, com exceção do subgrupo *Localização*. É responsável por realizar o CRUD de usuários e armazenar as informações dos mesmos. Também é responsável por armazenar informações de contexto dos usuários, referentes ao histórico de uso dos mesmos do sistema;
- **Localizador:** serviço que incorpora o grupo de *features Localização*, é responsável por intermediar a comunicação entre o sistema e os sensores IoT distribuídos pelo ambiente de operação, com o objetivo exclusivo de localizar os usuários.

Ainda há o *Orquestrador*, serviço no topo da hierarquia da MTTPL, incorpora a *feature MTTPL*, descendo pela árvore de *features* até as características *Informação de Contexto* e *Guia do Museu*. É responsável por expor o sistema todo aos usuários, bem como por implementar as funcionalidades providas pelos serviços presentes. O Orquestrador também é responsável por implementar o produto final efetivamente, arranjando o uso de cada funcionalidade do sistema formando os diferentes fluxos e casos de uso do sistema.

O Orquestrador é composto de dois componentes distintos: o *backend*, que implementa a orquestração dos serviços apresentados, e o *Aplicativo Móvel*, aplicação com o objetivo principal de exibir as interfaces gráficas do sistema aos usuários. O Aplicativo Móvel também incorpora *features* da MTTPL, sendo especificamente o grupo *Sensor de Smartphone*. Esta decisão baseia-se no fato de que tal aplicação já estaria em execução no dispositivo de onde os sensores dos *smartphones* seriam lidos. Assim, o sistema pode ser simplificado ao delegar esta função ao Aplicativo Móvel, e remover a necessidade de mais um serviço para a MTTPL.

É importante notar que o nome do componente *Aplicativo Móvel* indica que a visualização do sistema será implementada através de um aplicativo, ao invés de um portal *web* ou uma aplicação para computadores de mesa. Isto se deve ao fato da MTTPL mapear *smartphones* como o tipo de dispositivo que os usuários terão à mão para acessar o sistema. Assim, um aplicativo é a solução mais ideal, para utilizar mais facilmente da mobilidade dos usuários, bem como da maior facilidade em implementar a leitura dos sensores dos dispositivos.

4.7.2 Arquitetura Geral

Tendo a lista de serviços instanciáveis para a MTTPL, ainda se faz necessário entender como os mesmos se relacionam, para entender perfeitamente o papel desempenhado por cada serviço do sistema.

As relações são elaboradas com base nas atribuições dadas a cada sistema. Sendo assim, foram definidas as seguintes relações entre os serviços:

- o *Backend* consulta periodicamente o *Broker de Sensores* para atualizar os dados sobre o ambiente de operação do sistema;
- o *Backend* sempre consulta o *Provedor de Conteúdo* para recuperar as informações de itens do museu;
- o *Backend* sempre consulta o *Mapeador de Rotas* para saber quais rotas existem no sistema, e qual orientação de navegação deve dar ao usuário;
- o *Backend* sempre deve consultar o *Recomendador de Itens*, quando estiver instanciado, para saber se existem itens do museu a serem recomendados para o usuário, assim que consultar o *Mapeador de Rotas* por novas rotas e orientações;

- o *Recomendador de Itens* sempre consulta o *Provedor de Conteúdo* para recuperar os dados dos itens do museu, e aplicar métodos de decisão utilizando dados sobre os usuários, recebidos previamente do *Backend*;
- o *Backend* consulta o *Editor de Conteúdos*, quando instanciado, para realizar os procedimentos de cadastro de novos itens no sistema, por parte dos usuários;
- o *Editor de Conteúdos* sempre consulta o *Provedor de Conteúdo* para recuperar as informações de itens sendo editados, ou para salvar/excluir itens de acordo com a ação do usuário. O Editor de Conteúdos também deve ser responsável por controlar quais ações cada usuário tem autorização de realizar;
- tanto o *Backend* quanto o *Aplicativo Móvel* consultam sempre o *Provedor de Usuários* para atualizar o perfil dos usuários sempre que necessário, bem como recuperar seus dados e realizar tarefas de autenticação e sessão;
- quando o *Localizador* tiver que ser instanciado, ele deve ser implementado como parte do *Aplicativo Móvel*. Isto se deve ao fato de o Localizador utilizar tecnologias de comunicação e sensoriamento presentes no dispositivo móvel onde o aplicativo estará executando, e, portanto, seguindo o mesmo raciocínio utilizado na decisão de incorporar o grupo de *features Sensor de Smartphone* no Aplicativo Móvel;
- o *Aplicativo Móvel* consulta o *Provedor de Avaliações*, quando instanciado, para obter atividades a serem enviadas aos usuários, a fim de avaliar o conhecimento obtido durante a visita ao museu;
- por fim, o *Aplicativo Móvel* sempre consulta o *Backend* para obter os dados do sistema fornecido pelos outros serviços que o Aplicativo Móvel não tem acesso.

Desta forma, a lista de relações, apresentada acima, é ilustrada na [Figura 14](#). As linhas tracejadas indicam *dependências*, em que o sentido da seta aponta para o serviço acessado (se A aponta para B, então A depende de B, e, portanto, acessa suas APIs para executar as operações necessárias). É importante ressaltar que, de acordo com o que foi apresentado no [Capítulo 2](#), essas relações existem na forma de SLAs e usos de APIs REST.

Os serviços também são apresentados como *mandatórios* ou *opcionais*, indicando a obrigatoriedade de sua presença no ecossistema da MTTPL. Por fim, a relação entre o *Aplicativo Móvel* e o *Localizador* é diferente dos demais, dado que o segundo faz parte do primeiro. Isto significa que o Localizador é implementado dentro do Aplicativo Móvel, e, portanto, mudanças nas características do Localizador influenciam diretamente na organização dos *assets* do Aplicativo Móvel.

Figura 14 – Organização dos serviços da MTTPL.

4.8 Classes-Base

Como último passo na definição da MTTPL, resta definir como será a organização do código-fonte dos serviços que compõem a LPOS. Antes de prosseguir, é válido notar que os diagramas de classes aqui apresentados seguem a metodologia PLUS, e, portanto, servem apenas como um *guia* de como cada serviço deve ser implementado. Isso significa que os diagramas podem ser alterados caso o processo de engenharia de aplicação mostre tal necessidade.

4.8.1 Entidades Comuns

Como a MTTPL define um conjunto de serviços, todos os sistemas deste tipo necessitam de um *RestController*. Esta entidade é responsável por implementar as interfaces HTTP expostas pelo serviço. O *RestController* também pode ser interpretado como uma representação em baixo nível do SLA de um serviço.

4.8.2 Broker de Sensores

Este serviço agrega a leitura de diferentes tipos de sensores, então seu diagrama de classes consiste, basicamente, em uma entidade em nível mais alto representando a leitura dos sensores, e as devidas especializações. Sendo assim, este serviço deve implementar um *SensorReader*, representando o grupo de *features* *Sensores de Ambiente*, e as respectivas heranças, de forma a definir as possíveis leituras de diferentes tipos de sensores.

O modelo de classes do Broker de Sensores, com as informações citadas acima, pode ser visto na [Figura 15](#).

Figura 15 – Classes base do Broker de Sensores.

4.8.3 Recomendador de Itens

Este serviço oferece um maior grau de liberdade acerca de sua implementação, dado que nenhum aspecto sobre sistemas de recomendação foi mapeado para o mesmo. Isto se deve ao fato de sistemas de recomendação serem um domínio à parte de aplicações, e, a fim de simplificar o escopo deste trabalho, as características mais específicas deste tipo de sistema não foram levantadas dentre os requisitos da MTPL.

Assim, de acordo com o mapeamento feito visando referenciar este tipo de aplicação como um sistema externo, é necessária apenas uma entidade *ItemRecommender*, responsável por implementar as decisões sobre recomendação de itens. No entanto, este serviço também precisa de outra entidade, derivada da necessidade de integrar-se com o *Provedor de Conteúdo*. Para a entidade que lidará com essa integração, deu-se o nome *ItemBroker*.

O modelo de classes do Recomendador de Itens, com as informações citadas acima, pode ser visto na [Figura 16](#).

Figura 16 – Classes base do Recomendador de Itens.

4.8.4 Provedor de Conteúdo

Este serviço apresenta funcionalidades simples, tendo a responsabilidade de lidar apenas do *Create, Read, Update and Delete* (CRUD) de itens do museu. Desta forma, o Provedor de Conteúdo tem a principal responsabilidade de persistir os dados dos itens num banco de dados, e fazer a comunicação com o mesmo. Para lidar com isso, o padrão *Data Access Object* foi adotado (BAELDUNG, 2018). Além disso, para facilitar a representação dos dados, o padrão *Data Transfer Object* também foi adotado (SUN, 2002). Assim, este serviço possui duas entidades: *ItemDao* e *ItemDto*.

Para lidar com as mídias de cada *ItemDto*, que pode ter mais de uma mídia anexada, criou-se uma entidade *Media*. Além disso, como todo *ItemDto* tem uma descrição, optou-se por adicionar um campo *description*, do tipo *Media*. Isso é válido porque a entidade *Media* deve ser capaz de suportar texto como tipo de mídia.

O modelo de classes do Provedor de Conteúdo, com as informações citadas acima, pode ser visto na Figura 17.

Figura 17 – Classes base do Provedor de Conteúdo.

4.8.5 Editor de Conteúdos

Este serviço lida com a edição dos itens do museu, fazendo a validação dos dados fornecidos. Quando há a *feature Autoria por Usuários* instanciada, o serviço também deve realizar o controle de acesso para certas operações. Assim, o serviço tem uma entidade *ItemEditor* e uma entidade *EditorPermissionHandler*. Este serviço também precisa de um *ItemBroker*.

O modelo de classes do Editor de Conteúdos, com as informações citadas acima, pode ser visto na Figura 18.

Figura 18 – Classes base do Editor de Conteúdos.

4.8.6 Mapeador de Rotas

Este serviço deve ser capaz de gerar a rota até um item selecionado pelo usuário. De acordo com a escolha das *features* da MTTPL, também deve ser capaz de sugerir rotas ou de decidir automaticamente a rota pela qual o usuário deve seguir. Assim, optou-se por criar uma entidade *RoutePlanner*, com os campos *destination* e *pathList*. O campo *destination* é uma representação do destino final do caminho, que pode ser, por exemplo, a localização do item desejado. Já o campo *pathList* representa a lista de trechos do trajeto que o usuário deve percorrer para chegar até o item desejado.

Para lidar com a sugestão ou automatização das rotas, a entidade *RoutePlanner* recebeu duas especializações: *RouteDecider* e *RouteRecommender*. O modelo de classes do Mapeador de Rotas, com as informações citadas acima, pode ser visto na [Figura 19](#).

Figura 19 – Classes base do Mapeador de Rotas.

4.8.7 Provedor de Usuários

Este serviço é semelhante ao *Provedor de Conteúdo*. No entanto, a representação das entidades são para os usuários do sistema, portanto as entidades de representação são nomeadas *UserDao* e *UserDto*. Além disso, uma entidade *Interest* é necessária para mapear os interesses dos usuários, bem como uma entidade *Skill* é necessária para mapear os conhecimentos.

A entidade *Skill* recebeu um campo *isPrevious*, para representar se aquele conhecimento que um dado usuário possui foi adquirido antes de começar a utilizar o sistema. O modelo de classes do Provedor de Usuários, com as informações citadas acima, pode ser visto na [Figura 20](#).

Figura 20 – Classes base do Provedor de Usuários.

4.8.8 Provedor de Avaliações

Este serviço é responsável por fornecer as avaliações de conhecimento realizadas pelo sistema. Para isso, uma entidade *AssessmentPlanner* foi adicionada, para elaborar o questionário de avaliação com as perguntas consideradas mais relevantes.

Além disso, é recomendável que haja um banco de questões armazenado no banco de dados. Por isso, foram criadas duas entidades representacionais para tal: *QuestionDao* e *QuestionDto*. O modelo de classes do Provedor de Avaliações, com as informações citadas acima, pode ser visto na [Figura 21](#).

4.8.9 Localizador

O Localizador faz parte da implementação do *Aplicativo Móvel*, não sendo, portanto, um serviço. Graças a esta característica, o Localizador não precisa de um *RestController*. Além disso, como este componente trata da leitura de sensores capazes de localizar o usuário, ele tem uma estrutura semelhante ao *Broker de Sensores*. Assim, também foi utilizada a entidade *SensorReader*, com as devidas especializações: *BeaconReader*, *RFIDSensorReader* e *GPSReader*.

Figura 21 – Classes base do Provedor de Avaliações.

O modelo de classes do Localizador, com as informações citadas acima, pode ser visto na [Figura 22](#).

Figura 22 – Classes base do Localizador.

4.8.10 Backend

O *Backend*, conforme dito anteriormente, tem a função de orquestrar os serviços conectados a ele, de forma a prover as funcionalidades para o aplicativo. Além disso, o Backend também é responsável por supervisionar a execução completa de uma operação do sistema que dependa de mais de um serviço para ser executada. Por isso, ele precisa de uma entidade *GenericBroker* que implemente os métodos necessários a todos os *brokers* do sistema, bem como uma especialização para cada serviço. São as especializações: *EditorBroker*, *UserBroker*, *SensorBroker*, *ItemBroker* e *MapBroker*.

Além disso, este serviço também precisa de uma forma para armazenar as referências aos serviços que utilizará. Por isso, uma entidade *ServiceAddressRepository* também é necessária. O modelo de classes do Backend, com as informações citadas acima, pode ser visto na [Figura 23](#).

Figura 23 – Classes base do Backend.

4.8.11 Aplicativo Móvel

O aplicativo tem a responsabilidade de coletar dados dos sensores de ambientação do dispositivo móvel onde opera. Por isso, necessita de uma entidade *SensorReader*, e as especializações dos sensores mapeados: *ThermometerReader*, *GyroscopeReader* e *AccelerometerReader*. É importante ressaltar que o Aplicativo Móvel também deve implementar o *Localizador*, quando este for instanciado. Portanto, o *SensorReader* do aplicativo deve ser o mesmo apresentado no modelo de classes do Localizador. O aplicativo também precisa de um *AssessmentBroker* para recuperar avaliações de conhecimento do *Provedor de Avaliações*.

Um dos detalhes mais importantes do Aplicativo Móvel é a ausência de uma entidade *RestController*. Como o aplicativo acessa APIs ao invés de expor, não há propósito em manter esta entidade nesta parte do sistema. Por este motivo, esta entidade não aparece aqui. Por fim, uma entidade *UI* é necessária para exibir as interfaces gráficas do sistema aos usuários, bem como uma entidade *App* para habilitar a comunicação entre os *brokers* do aplicativo e as interfaces gráficas.

O modelo de classes do Aplicativo Móvel, com as informações citadas acima, pode ser visto na [Figura 24](#).

4.9 Considerações Finais

Neste capítulo, foi apresentada a MTTPL, LPOS projetada para auxiliar no desenvolvimento e reúso de sistemas de *software* voltados para o auxílio à visita de museus. A MTTPL utiliza informações coletadas a partir de estudos selecionados num SM feito previamente, utilizando tais dados para extrair características e variabilidades. A partir de tais características,

Figura 24 – Classes base do Aplicativo Móvel.

foram definidos os modelos de caso de uso da MTTPL, mapeando seus possíveis usos, bem como o modelo conceitual e modelo de *features*. Por fim, foram listados seus serviços e as relações entre estes foram mapeadas.

A partir dos artefatos apresentados neste Capítulo, é possível criar uma nova instância da MTTPL, seguindo os conceitos de LPOS e implementando as características apresentadas. Os modelos de classes apresentados servem como uma base sólida para a implementação do sistema, além de serem um mapeamento direto entre as *features* da MTTPL e os *assets* necessários para iniciar o processo de instanciação da LPOS.

SMARTICMC

5.1 Descrição Geral

Para que a MTTPL seja uma proposta válida, é necessário avaliar se a modelagem proposta traz benefícios práticos para a instanciación de novos produtos. No entanto, a validação de uma LPS é extremamente complexa, dado que é necessário validar cada combinação de características, bem como os produtos gerados a partir de cada combinação. Como o escopo deste trabalho é limitado pelo tempo disponível para a realização do mesmo, não é possível validar a MTTPL completamente.

Porém, algumas verificações de menor porte podem ser realizadas, buscando inferir qual é o resultado do processo de validação. Dentre essas verificações, encontra-se a *prova de conceito*, centrada na proposta de instanciar um produto da MTTPL e avaliar se ele incorpora as qualidades oferecidas pela linha de produtos, além de ser um sistema que se comporta conforme o esperado.

No ICMC/USP, há um museu a céu aberto, o Museu da Fauna e da Flora do ICMC, cuja facilidade de visualização para os visitantes do instituto é reduzida. Assim, identificou-se a possibilidade de criar um produto capaz de atender às necessidades deste museu, utilizando a MTTPL como ponto de partida e o produto instanciado como prova de conceito da mesma. A este produto gerado a partir da MTTPL, deu-se o nome de *SmartICMC*.

O SmartICMC atende aos requisitos do domínio de museus virtuais oferecido pela MTTPL, já que fornece suporte à visitação de um museu onde atividades de aprendizado podem ser conduzidas, ao considerar que um de seus objetivos é permitir que os visitantes tenham maior contato com diferentes espécimes de árvores. Assim, os visitantes podem observar de perto os detalhes de cada uma destas espécies de árvores, bem como ler algumas informações presentes em algumas placas distribuídas ao longo do museu. Portanto, ao visitar o museu por meio do SmartICMC, uma pessoa pode adquirir novos conhecimentos em botânica.

Também é útil para o Museu da Fauna e da Flora que o SmartICMC seja capaz de operar com dispositivos IoT, dado que o ambiente de operação do sistema se encontra a céu aberto, o que faz com que os dispositivos a se comunicarem sejam independentes de uma infra-estrutura física. Como os dispositivos IoT são capazes de operar com eficiência e independência em sistemas baseados em computação em nuvem, representam recursos satisfatórios para o SmartICMC e oferecendo mais características semelhantes às que a MTTPL oferece.

No entanto, o uso de IoT que o SmartICMC requer possui menor grau de complexidade, exigindo, dentre seus requisitos, que o sistema esteja ciente apenas da posição dos usuários dentro do museu. Assim, pode-se utilizar os *smartphones* dos visitantes como dispositivo IoT, e seus sensores internos para captar dados referentes à sua posição e orientação. Com a leitura do GPS e giroscópio, o dispositivo é capaz de inferir em que local do museu o visitante se encontra, e para onde o mesmo está se dirigindo. Assim, o sistema pode utilizar estes dados para orientar o usuário sobre a direção que deve seguir, adicionando a ciência de contexto necessária para tornar a navegação mais eficiente.

Além disso, o SmartICMC também foi utilizado como um exemplo de como novas instâncias da MTTPL podem ser criadas, a partir da escolha de diferentes linguagens e tecnologias específicas para a etapa de implementação dos elementos da LPOS.

5.2 Objetivo

O objetivo do SmartICMC é atender à demanda existente no Museu da Fauna e da Flora de ampliar a visibilidade e reconhecimento do museu através da facilitação à visitação dos espécimes presentes no mesmo. O não cumprimento deste objetivo inviabiliza, de imediato, o produto instanciado. No entanto, devido à inviabilidade de rastrear os animais que passam pelo Museu por tratarem-se principalmente de aves, o contexto de operação do SmartICMC será restrito apenas à flora ali presente. Futuramente, havendo disponibilidade de tecnologias que possam detectar a localização de aves e outros animais do museu (por exemplo, chips implantados nos animais ou sistemas de reconhecimento de imagens), o SmartICMC poderá passar por uma evolução para atender tais requisitos.

Utilizando este objetivo, é possível traçar parâmetros de avaliação para verificar a correteza do produto instanciado a partir da MTTPL. Obtendo êxito em cumprir este objetivo, a MTTPL pode ser caracterizada como utilizável para este domínio de aplicações.

Para o SmartICMC, os seguintes objetivos são mandatórios:

- Facilitar a navegação dos visitantes pelo museu;
- Utilizar o contexto de uso dos visitantes para facilitar ações no sistema.

Devido à simplicidade do SmartICMC, pode-se considerar que a prova de conceito a ser gerada para a MTTPL baseia-se em um Produto Mínimo Viável, ou *Minimum Viable Product* (MVP), de forma que seja instanciado um sistema que tenha algum valor real agregado, e que demande o menor esforço de trabalho necessário ao mesmo tempo (MOOGK, 2012).

5.3 Engenharia do Sistema

Conforme dito anteriormente, o SmartICMC foi planejado para ser uma LPOS instanciada a partir da MTTPL. Portanto, para definir os artefatos a serem instanciados do SmartICMC, é necessário realizar o processo de engenharia de aplicação (GOMAA, 2005) sobre a MTTPL, para derivar os artefatos específicos do SmartICMC, a partir dos artefatos da MTTPL, que são apresentados no Capítulo 4.

5.3.1 Features

O primeiro passo na definição do SmartICMC é selecionar as *features* da MTTPL que serão instanciadas inicialmente. Para isso, foi necessário analisar as necessidades do SmartICMC, e selecionar o conjunto de características da MTTPL que as melhor satisfaçam. Como os próprios desenvolvedores do SmartICMC também são os *stakeholders* do projeto, bastou apenas selecionar as características desejadas, de acordo com o que era esperado do sistema.

O modelo de *features* do SmartICMC pode ser visto na Figura 25, em que as características selecionadas podem ser vistas coloridas em tom alaranjado.

5.3.2 Serviços

A próxima etapa consiste em definir, a partir das *features* selecionadas, quais são os serviços que deverão ser instanciados para o SmartICMC. Para tal, basta observar as características selecionadas, e comparar com os serviços listados pela MTTPL, a fim de identificar quais são os serviços abrangidos pelo modelo de *features* instanciado para o SmartICMC.

De acordo com as características selecionadas, visíveis na Figura 25, e retomando os serviços listados da MTTPL no Capítulo 4, sabe-se que os serviços a serem instanciados para o SmartICMC são: *Provedor de Conteúdo*, *Localizador*, *Mapeador de Rotas* e *Provedor de Usuários*. Também é sabido que ambos *Backend* e *Aplicativo Móvel* são partes do *Orquestrador*, e que, portanto, suas presenças são mandatórias no sistema final.

Assim, o modelo organizacional dos serviços do SmartICMC pode ser visto na Figura 26, com os serviços a serem instanciados coloridos em tom alaranjado.

Figura 25 – Modelo de *features* do SmartICMC, a partir do modelo da MTTPL.

5.3.3 Classes do SmartICMC

Sabendo quais são os serviços a serem instanciados, ainda resta ajustar os diagramas de classe fornecidos pela MTTPL, a fim de remover as classes que não são necessárias. Para realizar este ajuste, é necessário comparar cada serviço a ser instanciado com as diferenças entre as *features* que o compõem, no que se refere à MTTPL e SmartICMC. Em síntese, as classes e campos que representam características da MTTPL não selecionados para o SmartICMC devem ser removidos.

Figura 26 – Serviços do SmartICMC, a partir do modelo definido pela MTTPL.

O resultado deste ajuste pode ser visto nas Figuras 27 a 31. É importante notar que o *Provedor de Conteúdo*, visto na Figura 29, não sofreu alterações devido à ausência de algumas *features* presentes em sua modelagem original. Isso se deve ao fato de como os diferentes tipos de mídias de conteúdo foram mapeadas para dentro do serviço, de forma que o mesmo seja capaz de prover tais conteúdos não dependendo do formato dos mesmos.

Outro aspecto importante de ser notado é a morfologia do Aplicativo Móvel, juntamente com a ausência do Localizador. Isto se deve ao fato do Localizador ser implementado pelo Aplicativo Móvel, conforme descrito no Capítulo 4. Por isso, o Aplicativo Móvel apresenta as especializadas do *SensorReader* introduzidas pelo Localizador instanciado.

Figura 27 – Aplicativo Móvel do SmartICMC.

Figura 28 – Mapeador de Rotas do SmartICMC.

Figura 29 – Provedor de Conteúdo do SmartICMC.

Figura 30 – Backend do SmartICMC.

Figura 31 – Provedor de Usuários do SmartICMC.

5.3.4 Tecnologias de Implementação

O próximo passo na engenharia de aplicação do SmartICMC é ajustar os modelos de classes derivados da MTTPL, a fim de gerar modelos próximos dos diagramas de classes finais de cada serviço do SmartICMC. No entanto, como a escolha das tecnologias usadas na implementação destes serviços pode impactar a morfologia final dos diagramas de classes, se faz necessária a definição de tais tecnologias antes de prosseguir com a finalização dos diagramas.

Os serviços podem ser tratados como pequenas aplicações sem interfaces gráficas, com objetivos precisamente delineados e escopo bem definido. Portanto, tecnologias que colaborem bem com estas características são desejáveis sobre outras que possam dificultar o desenvolvimento dos serviços. Além disso, qualquer tecnologia que apresente formas prontas de desenvolver serviços *web*, com fácil integração com bancos de dados, possuem preferência de escolha.

5.3.4.1 Java

Considerando as características dos serviços do SmartICMC e o que é desejável para as tecnologias a serem utilizadas, optou-se pelo Java como linguagem de desenvolvimento (ORACLE, 2019a). Java é uma linguagem extremamente amadurecida atualmente, com milhares de bibliotecas disponíveis para facilitar e acelerar o desenvolvimento de novos sistemas. Além disso, existem diversas ferramentas de versionamento de bibliotecas para sistemas Java que são amplamente usadas atualmente, como *Maven* e *Gradle*. Portanto, com a opção pelo Java, é certo de que haja bibliotecas que atendam às demandas do SmartICMC, acelerando e facilitando o desenvolvimento dos seus serviços. Além disso, por ser uma linguagem com alto grau de maturidade, Java possui uma comunidade de desenvolvedores mais antiga, o que significa que é mais provável que outros desenvolvedores tenham enfrentado os mesmos problemas que possam surgir durante o desenvolvimento de qualquer serviço do SmartICMC, portanto sendo mais fácil

superar tais dificuldades.

5.3.4.2 Spring Boot

Com a opção pelo Java, também foi possível escolher a biblioteca *Spring Boot* (PIVOTAL, 2019), que possibilita a implementação de sistemas RESTful rapidamente, sem precisar configurar *web services* nos mínimos detalhes. A *Spring Boot* permite que as APIs de uma aplicação sejam codificadas apenas com base em suas declarações, de forma que o código-fonte seja uma tradução próxima do SLA de um serviço. Assim, com o uso desta biblioteca, o desenvolvimento das APIs dos serviços do SmartICMC seria simples e rápido, além de documentar tais APIs em um formato legível para outros desenvolvedores.

Outro padrão que a *Spring Boot* confere a toda a aplicação em que é utilizada é o *Singleton*. A biblioteca implementa entidades a serem compartilhadas pelos diferentes objetos do sistema sob o nome de *beans*, e a instanciação destes é gerenciada pela própria biblioteca, que força que sejam únicos. Ao utilizar tais *beans*, a *Spring Boot* facilita a centralização de configurações do sistema, bem como oferece formas de distribuí-los através da aplicação por meio de mecanismos de injeção de dependências, ou *Dependency Injection* (DI) (YANG; TEMPERO; MELTON, 2008).

5.3.4.3 MySQL

Além da escolha de Java como linguagem, optou-se por utilizar o MySQL (ORACLE, 2019b) como banco de dados. Esta escolha foi motivada pela rapidez com que o banco de dados pode ser criado e conectado ao sistema. Após a aquisição pela Oracle, o MySQL ganhou uma interface gráfica mais intuitiva, facilitando tarefas de configuração como a criação de credenciais de acesso e execução de *scripts* de criação das tabelas do banco.

Além das características acima, é importante notar que o MySQL, em sua versão gratuita, oferece ferramentas de gerenciamento suficientes para sistemas que façam um uso mais simplificado do banco de dados, e não precisem de aspectos mais avançados como criptografia de dados ou *backups* periódicos. Portanto, para o SmartICMC, o MySQL é uma opção que atende às demandas do sistema.

5.3.4.4 jOOQ

jOOQ é uma biblioteca específica para sistemas Java, que se baseia no padrão Mapeador Objeto Relacional, ou *Object Relational Mapper* (ORM) (GEEKERY, 2009). O uso desta biblioteca facilita o desenvolvimento de aplicações que tenham conexão com bancos de dados, pois a biblioteca é quem realiza a conexão com os bancos e gerencia as *queries* enviadas bem como os resultados recebidos. Além disso, a *jOOQ* gera automaticamente uma classe *Data Access Object* (DAO) e uma *Data Transfer Object* (DTO) para cada tabela presente no banco de dados, assim permitindo que o desenvolvedor da aplicação trate a comunicação com o banco de

dados como se fosse uma mera invocação de métodos de objetos. Obviamente, o desenvolvedor também pode implementar outras DAOs que realizem consultas mais complexas ao banco de dados, utilizando os DAOs gerados pelo jOOQ para isto.

5.3.4.5 JavaScript

Para o desenvolvimento do *Aplicativo Móvel*, optou-se por utilizar o *JavaScript* como linguagem de desenvolvimento. Esta linguagem é interpretada pelos navegadores desde versões mais antigas do *Internet Explorer*, e, portanto, já é utilizada para o desenvolvimento de páginas *web* há vários anos. A Fundação Mozilla mantém uma documentação atualizada da linguagem, bem como uma lista de referência sobre o suporte que cada navegador tem às APIs da linguagem (MOZILLA, 2019).

No entanto, o principal motivo de optar pelo JavaScript são as bibliotecas que permitem o desenvolvimento de uma *Single Page Application* (SPA). Este tipo de aplicação é popular atualmente, graças a bibliotecas como Angular (ANGULAR, 2019) e Vue.js (VUE.JS, 2019), principalmente pelo desempenho que aplicações *web* desenvolvidas nestes moldes apresentam sobre aplicações tradicionais.

5.3.4.6 React

O *React* é uma biblioteca para desenvolvimento de SPAs, criada e mantida pelo *Facebook* (FACEBOOK, 2019). Esta biblioteca é dita ser a com maior desempenho dentre todas as bibliotecas de desenvolvimento de SPAs atualmente, segundo os engenheiros responsáveis pela manutenção da biblioteca.

Além disso, o React apresenta uma curva de aprendizado mais custosa para iniciantes, porém oferecendo maiores recompensas conforme o desenvolvedor acumula conhecimento e experiência no seu uso. Esta biblioteca se baseia no princípio de *componentização* da aplicação: todos os elementos de uma interface gráfica são divididos em componentes, com responsabilidades precisamente delineadas. Assim, uma boa componentização dos elementos gráficos confere graus mais elevados de reuso dos componentes da aplicação. Outro fator que contribui para o alto grau de reuso dos componentes React é a ausência de regras de negócio do sistema, que deve sempre ser evitada (MENDONCA, 2018; ABIRACHED, 2017).

Somado ao alto grau de reuso de código-fonte, pelo fato de aplicações React serem desenvolvidas como uma SPA, elas podem ser implantadas em ambientes de produção como páginas *estáticas*, assim como portais *web* monolíticos. Assim, apesar de serem desenvolvidos utilizando tecnologias mais modernas, portais *web* desenvolvidos com React também são compatíveis com servidores de distribuição mais antigos.

5.3.4.7 Webpack e Babel

O *Webpack* é a principal ferramenta de versionamento de bibliotecas e empacotamento utilizada em projetos desenvolvidos utilizando JavaScript. Além disso, também é a ferramenta mais utilizada em projetos com React, sendo recomendada pela equipe do Facebook. O uso do Webpack traz facilidade no gerenciamento de bibliotecas e no processo de geração dos pacotes para implantação, uma vez que é possível configurar este processo de empacotamento através de *scripts*.

O uso do Webpack também facilita a utilização do *Babel*, biblioteca JavaScript que faz a tradução de padrões de linguagem mais novos para padrões mais antigos, que os navegadores sejam capazes de processar. Além disso, o Babel também oferece a possibilidade de diminuir o tamanho dos arquivos gerados, realizando um processo chamado de *minificação*, do inglês *minify*, bem como transformar o código-fonte da aplicação numa versão humanamente ilegível e, assim, escondendo a lógica da aplicação, através de um processo chamado de *uglificação*, do inglês *uglify*. O uso do Babel permite que o desenvolvedor utilize as ferramentas e técnicas mais poderosas que o JavaScript oferece, sem se preocupar com a compatibilidade com o navegador.

5.3.4.8 Plataformas de Implantação

Para facilitar o desenvolvimento do SmartICMC, é desejável utilizar uma plataforma que ofereça uma fácil implantação de aplicações. É desejável utilizar três plataformas distintas: uma para o Aplicativo Móvel, uma para o banco de dados, e outra para o Backend e os outros serviços do SmartICMC.

Para o banco de dados, utilizou-se um banco MySQL instalado no servidor da USP, mais precisamente dentro do servidor do Laboratório de Engenharia de Software (LabES). Para o Aplicativo Móvel, a plataforma Netlify ([NETLIFY, 2019](#)) provou-se suficiente para atender às demandas do sistema. Por fim, para os serviços, utilizou-se o Heroku ([SALESFORCE, 2019](#)).

5.4 Organização do Sistema

Tendo os artefatos gerados durante a Engenharia de Aplicação, descrita na seção anterior, em mãos, resta definir a morfologia de cada sistema constituinte do SmartICMC antes de, finalmente, elaborar os códigos-fonte. Para isso, é necessário comparar os requisitos do SmartICMC com todos os artefatos gerados, e incorporar à modelagem de cada subsistema os requisitos faltantes, a fim de realizar os últimos ajustes dos modelos da MTPL para o domínio do SmartICMC.

A seguir, são apresentados os diagramas de classes de cada aplicação do SmartICMC, referentes às alterações feitas que impactam os modelos obtidos até o momento. É importante esclarecer que aqui são apresentadas e comentadas somente as mudanças que afetam a modela-

gem apresentada, sem cobrir a adição de novas classes necessárias para cada sistema funcionar conforme o esperado.

5.4.1 Provedor de Usuários

Para este serviço, é necessário definir quais informações o sistema irá armazenar sobre os usuários. No caso do SmartICMC, são o nome, o *e-mail* e a senha de acesso ao sistema. Portanto, além de permitir recuperar os dados do usuário, o Provedor de Usuários também deve tratar os casos de *login* e *logout* do usuário, bem como a alteração de algum dado por parte do usuário. O resultado deste mapeamento pode ser visto na Figura 32.

Figura 32 – Provedor de Usuários do SmartICMC após a modelagem dos requisitos faltantes.

5.4.2 Provedor de Conteúdo

Para este serviço, foi adicionada apenas uma API, para recuperar um item específico do museu. Além disso, foi adicionado um campo na classe *ItemDto* para armazenar o nome de cada árvore. O resultado pode ser visto na Figura 33.

5.4.3 Mapeador de Rotas

A classe *RoutePlanner* possuía, originalmente, duas possíveis especializações, para o caso do produto instanciado implementar a recomendação ou automatização das rotas que os visitantes devem seguir. No entanto, para o SmartICMC, as *features* que representam tais características, respectivamente *Sugestão de Rotas* e *Rotas Automatizadas*, não foram instanciadas. Assim, a instância deste serviço para o SmartICMC não possui tais especializações.

Além disso, foram adicionadas APIs para recuperar um nó do mapa, bem como a lista de nós. Estas APIs definem que o mapa do museu será modelado como um grafo, cujos nós indicam

Figura 33 – Provedor de Conteúdo do SmartICMC após a modelagem dos requisitos faltantes.

os pontos de conexão entre os caminhos do museu. Além disso, o grafo do mapa também terá nós desconexos do grafo, indicando a localização das árvores do museu. Isto foi adicionado por decisões de projeto em modelar o mapa do museu desta forma.

O resultado pode ser visto na Figura 34.

Figura 34 – Mapeador de Rotas do SmartICMC após a modelagem dos requisitos faltantes.

5.4.4 Backend

O Backend teve todas as APIs dos serviços incluídas nele, de forma a expor tais funcionalidades para o Aplicativo Móvel. Assim, o aplicativo precisa se comunicar apenas com o Backend, e este pode tratar os dados recebidos dos outros serviços para adequá-los ao SLA a que o Aplicativo Móvel obedece. Além disso, o endereço de cada serviço acessado pelo Backend foi incorporado no *ServiceAddressRepository*, de forma que a aplicação tenha uma referência única de tais endereços, utilizando o padrão *Singleton*, mencionado anteriormente.

Por fim, os *brokers* do Backend foram modelados em um padrão genérico, de forma que o *RestController* acesse as APIs de outros serviços utilizando os métodos expostos pelo *GenericBroker*, sem precisar de acesso aos métodos específicos de cada classe especialista. O resultado final pode ser visto na Figura 35.

Figura 35 – Backend do SmartICMC após a modelagem dos requisitos faltantes.

5.4.5 Aplicativo Móvel

Diversas customizações sobre o modelo de classes do Aplicativo Móvel tiveram de ser realizadas, a fim de ajustá-lo para as especificidades do projeto. De início, foram incluídas as informações de cada tipo de sensor de *smartphone* como campos de cada classe especialista. Além disso, um método comum também foi adicionado para capturar tais leituras e retorná-las.

A entidade *App* foi mantida na modelagem, pelo fato de existir em sistemas React. Ela é utilizada como ponto inicial de renderização da interface gráfica, adicionando todos os decoradores de componentes necessários e que tenham a responsabilidade de prover funcionalidades a todos os componentes do sistema.

Por fim, a entidade *UI* também foi mantida, mas por uma mera questão *representacional*. Ao invés de modelar todos os componentes das interfaces gráficas, uma única entidade é suficiente para representar todos, no caso da modelagem utilizada para o SmartICMC. Isto é possível pela mudança de relação do *SensorReader* com o *App*: como a UI passa a *utilizar* o *SensorReader*, *todos* os componentes do aplicativo passam a ter tal relação. Esta mudança significa que o *SensorReader* foi projetado como um componente de alta ordem, ou *High Order Component* (HOC): um componente decorador, que injeta propriedades no componente decorado. No caso do Aplicativo Móvel, o *SensorReader*, nesta modelagem, injetará as leituras dos sensores nos componentes que o utilizarem.

O resultado destas mudanças pode ser visto na Figura 36.

Figura 36 – Aplicativo Móvel do SmartICMC após a modelagem dos requisitos faltantes.

5.5 Código-fonte Derivado

É extremamente complexo de avaliar a corretude de uma LPS, devido à necessidade de explorar todas as combinações de *features* possíveis, a fim de verificar se cada um dos produtos instanciáveis é gerado corretamente. Assim, para o escopo deste trabalho, é inviável realizar a validação do SmartICMC.

No entanto, é possível realizar inspeções no código-fonte¹, buscando as classes e métodos que apresentam pontos de decisão onde a instanciação de *features* da MTTPL é capaz de alterar o comportamento da entidade em questão. Desta forma, para o escopo deste trabalho, foi tomada esta inspeção do código-fonte como suficiente para verificar se o que foi gerado reflete os requisitos do sistema.

5.5.1 Backend - RestController

Como o *Backend* tem um papel mais voltado à centralização da comunicação e das informações, a inspeção do código-fonte visa avaliar se o mesmo se comunica *apenas* com os sistemas instanciados pelo SmartICMC. Sendo assim, bastou verificar sua classe *RestController* a fim de verificar a lista de serviços conectados.

Na Figura 37, é possível ver que o *Backend* possui referência apenas para os *brokers* de conteúdo, mapa e usuários. Isto significa que o *Backend* se comunica apenas com os serviços *Mapeador de Rotas*, *Provedor de Conteúdos* e *Provedor de Usuários*, o que é esperado segundo as *features* da Figura 25.

¹ Códigos-fonte disponíveis em <https://gitlab.com/Fuzetaum/smarticmc-server> e <https://gitlab.com/Fuzetaum/smarticmc-app>

Figura 37 – *Brokers* do *Backend* instanciados pelo *RestController*.

```
public class RestController {
    @Autowired
    private GenericBroker backendItemBroker;

    @Autowired
    private GenericBroker backendMapBroker;

    @Autowired
    private UserBroker backendUserBroker;
}
```

5.5.2 Mapeador de Rotas - *RoutePlanner*

A inspeção do código-fonte da classe *RoutePlanner* visa verificar que as duas classes descartadas durante a engenharia de aplicação do SmartICMC, *RouteDecider* e *RouteRecommender*, não foram instanciadas. No entanto, visando prover um grau de reuso mais elevado, a classe *RoutePlanner* também deve ser capaz de incorporar código-fonte que permita não apenas instanciar tais classes, como utilizá-las.

Este é um caso onde espera-se que o código-fonte apresente *desvios de fluxo*, de forma que o comportamento dos métodos envolvidos e os resultados obtidos após suas execuções estejam de acordo com as *features* instanciadas pelo SmartICMC. O resultado pode ser visto na Figura 38, em que as *features* *Rotas Automatizadas* e *Sugestão de Rotas* não possuem implementação, conforme esperado.

Figura 38 – Código-fonte do método *getMapNodes* da classe *RoutePlanner*.

```
public List<MapNode> getMapNodes() {
    if (Features.ROTAS_AUTOMATIZADAS) return null;
    else if (Features.SUGESTAO_DE_ROTAS) return null;
    else
        return dslContext.selectFrom(MAP_NODE).fetch()
            .map(mapNode -> new MapNode(mapNode))
            .collect(Collectors.toList());
}
```

5.5.3 Provedor de Conteúdo - *ItemDao*

O *Provedor de Conteúdo* é um serviço que provê acesso aos itens do museu, basicamente. Desta forma, é esperado que este possua um funcionamento bastante simplificado, oferecendo interfaces para recuperar os itens do museu. No entanto, para o SmartICMC, um dos requisitos que deve ser tratado é o das curtidas, ou *likes*, associados a um item. Como os *likes* de um item

são apresentados apenas quando os dados de um item são exibidos, optou-se por calcular o total de *likes* de cada item quando seus dados são requisitados ao Provedor de Conteúdo.

O resultado pode ser visto na [Figura 39](#). Como um *like* é, por definição, um interesse de um usuário em um item, esta informação é mapeada via *feature Interesses*, e, portanto, o cálculo dos *likes* de um item deve estar associado à presença desta característica. Como há um desvio de fluxo condicionado à presença desta *feature*, é levantada mais uma evidência de que este serviço foi instanciado corretamente.

Figura 39 – Código-fonte do método *getItem* da classe *ItemDao*.

```
public ItemDto getItem(Integer id) {
    Item item = jooqItemDao.fetchById(i
    if (item != null) {
        Media description = jooqMed
        List<Media> mediaList = dsl
            .map(media
        ItemDto itemDto = new ItemD

        if (Features.INTERESSES) {
            Integer amountOfLik
                .fe
            itemDto.setAmountOf

        }

        return itemDto;
    }
}
```

5.5.4 Provedor de Usuários - RestController

Por fim, a avaliação do *Provedor de Usuários* visa buscar evidências de que entre todas as *features* do grupo *Perfil de Usuário*, excluindo o grupo *Localização* apenas a característica *Interesses* foi instanciada.

De acordo com a [Figura 40](#), não somente a *feature Interesses* teve funcionalidades implementadas para a mesma, como a característica *Aprendizado Adquirido* está presente, mas não apresentando algum tipo de código-fonte implementando suas funcionalidades. Assim, a instanciação desta última característica exige que um novo código-fonte seja incorporado, para implementá-la de acordo.

Figura 40 – Código-fonte do método *logoutUser* da classe *RestController*.

```
public void logoutUser(UserVisitDto userVisit) {  
    if (Features.INTERESSES && !Features.APRENDIZADO_ADQUIRIDO)  
        userProviderUserDao.logoutUser(userVisit.getUsername(), userVisit.getLiked());  
    else if (!Features.INTERESSES && Features.APRENDIZADO_ADQUIRIDO);  
    else if (Features.INTERESSES && Features.APRENDIZADO_ADQUIRIDO);  
    else  
        userProviderUserDao.logoutUser(userVisit.getUsername());  
}
```

5.6 Considerações Finais

Neste capítulo, foi apresentado o SmartICMC, sistema instanciado a partir da MTTPPL. Foram apresentados os artefatos desta LPOS para que a mesma seja descrita o suficiente, os quais foram instanciados segundo o processo de engenharia de aplicação seguido. Também foram apresentados trechos de código-fonte, exemplificando o resultado da geração dos *assets* finais do SmartICMC tomando os artefatos gerados como base para a geração do código.

Em uma análise preliminar, os resultados obtidos se mostram satisfatórios e o SmartICMC aparenta ter sido instanciado corretamente, considerando a necessidade de ser modelado como uma linha de produtos de *software*. No entanto, no Capítulo 6 é feita uma análise mais profunda acerca dos artefatos apresentados, visando verificar se o SmartICMC obedece efetivamente aos objetivos traçados, tanto para a MTTPPL quanto para o próprio sistema.

RESULTADOS

Apesar da instanciação do SmartICMC ter sido bem sucedida no que se refere a ter uma aplicação derivada no fim do processo, ainda não é sabido se essa instanciação serve como uma *prova de conceito* para validar a proposta da MTTPL. Isto se deve ao fato de ainda não ter sido feita uma análise dos artefatos e do código-fonte do SmartICMC, comparando-os com os artefatos da MTTPL. Além disso, o resultado obtido com o SmartICMC não foi avaliado segundo os objetivos traçados para a MTTPL.

6.1 *Assets* Implementados

Um dos primeiros resultados a serem observados a partir da instanciação realizada é o volume de *assets* reusáveis gerados, que podem ser reaproveitados em outros projetos que utilizem as mesmas tecnologias escolhidas para o SmartICMC.

Segundo a [Figura 25](#), foram instanciadas 16 das 36 *features* da MTTPL, excluindo a característica raiz *MTTPL*, que sempre será instanciada. Analogamente, ao analisar os modelos de classes dos serviços, 24 das 49 classes apresentadas foram geradas pelo SmartICMC. Isto significa que cerca de 50% dos *assets* reusáveis propostos pela MTTPL foram instanciados pelo SmartICMC, provendo uma base considerável de artefatos para serem reusados em futuros projetos. A proporção de *assets* reusáveis cresce se comparados apenas entre os serviços selecionados para instanciação, com apenas 32 classes totais e sendo 75% destas implementadas pelo SmartICMC.

Para verificar se os *assets* gerados poderiam ser utilizados no contexto do SmartICMC, testes funcionais rápidos e rasos foram feitos, verificando se a integração entre os serviços é correta mesmo com a presença de pontos de decisão decorrentes do modelo de *features*. Para ilustrar, as [Figuras 41a](#) e [41b](#) representam as duas principais telas do aplicativo, apresentando a forma como o aplicativo representa o mapa do museu e o cartão de informações de uma árvore,

respectivamente. A quantidade baixa de dados presente nas figuras se deve ao conjunto de dados de teste utilizados no momento do registro destas imagens.

Figura 41 – Capturas de tela do SmartICMC.

6.2 Classes

Conforme apresentado nos Capítulos 4 e 5, foram gerados diagramas de classes para o SmartICMC derivados dos modelos fornecidos pela MTTPL. Assim, é possível avaliar se a instanciação de tais classes foi feita corretamente, seguindo o modelo de *features* da LPOS para manter o mapeamento das características selecionadas.

Antes de qualquer análise, é importante notar a presença da classe *Features* no projeto. Esta entidade representa todo o conjunto de características do SmartICMC, incluindo aquelas que não foram instanciadas para esta primeira versão da linha de produtos. Como a linguagem de programação utilizada não oferece recursos para implementar diretivas de *compilação condicional* ou semelhantes, é necessário incluir no código-fonte pontos de decisão que modifiquem o comportamento dos métodos implementados. Por isso, a presença de uma entidade que sirva como referência única de consulta a respeito da presença de características da linha de produtos facilita a tradução destes comportamentos para código-fonte. As propriedades fornecidas pela classe *Features*, a nível de código-fonte, são retratadas pela Figura 42.

Analisando o *Backend* da MTTPL, visível na Figura 23 do Capítulo 4, e comparando-o com sua instanciação, visto na Figura 35 do Capítulo 5, é possível notar que as classes

Figura 42 – Características do SmartICMC, representadas em código-fonte através da classe *Features*.

```

public class Features {
    // Ciencia de Contexto
    public static final boolean SISTEMA_DE_RECOMENDACAO = false;
    // Informacao de Contexto
    public static final boolean PERFIL_DE_USUARIO = true;
    public static final boolean LOCALIZACAO = true;
    public static final boolean APRENDIZADO_PREVIO = false;
    public static final boolean APRENDIZADO_ADQUIRIDO = false;
    public static final boolean INTERESSES = true;
    public static final boolean RFID_DE_LOCAL = false;
    public static final boolean GPS = true;
    public static final boolean BEACON = false;
    public static final boolean AMBIENTE_DE_INSERTAO = false;
    public static final boolean RFID_DE_AMBIENTE = false;
    public static final boolean RASPBERRY_PI = false;
    public static final boolean ARDUINO = false;
    public static final boolean GIROSCOPIO = true;
    public static final boolean ACELEROMETRO = false;
    public static final boolean TERMOMETRO = false;
    // Guia do Museu
    // Visitacao Assistida
    public static final boolean ROTAS_AUTOMATIZADAS = false;
    public static final boolean SUGESTAO_DE_ROTAS = false;
    public static final boolean AVALIACAO_DE_CONHECIMENTO = false;
    public static final boolean ROTA_ATE_OBJETO = true;
    public static final boolean AUTORIA_DE_CONTEUDO = false;
    public static final boolean AUTORIA_POR_USUARIOS = false;
    public static final boolean SUGESTAO_DE_CONTEUDOS = false;
    // Midias de Conteudo
    public static final boolean TEXTO = true;
    public static final boolean IMAGEM = true;
    public static final boolean AUDIO = false;
    public static final boolean VIDEO = false;
}

```

EditorBroker e *SensorBroker* foram removidas. Como tais classes não são mapeadas para o SmartICMC devido à escolha das *features* não abranger tais entidades, pode-se considerar que a modelagem e instanciação do *Backend* foi realizada corretamente.

Outro serviço que pode ser analisado é o *Provedor de Usuários*, e a análise deste pode ser feita em maior profundidade. Fazendo as mesmas comparações dos diagramas de classe (Figura 20 do Capítulo 4, e Figura 32 do Capítulo 5), é possível notar a remoção da classe *Skill*, devido à não escolha das *features* *Aprendizado Prévio* e *Aprendizado Adquirido*. Isto indica um grau de correteza na instanciação deste serviço, porém é possível ir um passo além, dado

o código-fonte apresentado na [Figura 40](#). De acordo com as características do SmartICMC escolhidas para a instanciação de um novo produto, as *features* que compõem o grupo *Perfil de Usuário* podem aumentar ou não o perfil retornado pelo serviço, através das decisões contidas no método que avaliam a presença de tais características. Também é importante notar que a instanciação de uma característica não é necessariamente refletida em um único serviço, como pode ser visto nesta última figura.

6.3 Objetivos do SmartICMC

Conforme mencionado no Capítulo 5, o SmartICMC foi concebido com dois objetivos a serem seguidos estritamente:

- Facilitar a navegação dos visitantes pelo museu;
- Utilizar o contexto de uso dos visitantes para facilitar ações no sistema.

Estes dois objetivos foram cumpridos segundo a implementação do *Aplicativo Móvel*, que utiliza uma espécie de bússola para auxiliar os usuários a se orientarem pelo museu. Além disso, esta parte do sistema também utiliza o GPS do *smartphone* para implementar um sensor de proximidade, analisando constantemente a distância entre o visitante e a árvore do museu. Quando o usuário se aproxima o suficiente (a medida de distância utilizada para isto foi definida arbitrariamente) da árvore selecionada, o aplicativo automaticamente exibe o cartão informativo daquele item. Uma versão intermediária desta bússola, funcional mas com informações de depuração disponíveis na tela, pode ser vista na [Figura 43](#).

Devido a este comportamento, pode-se considerar que os dois objetivos traçados para o SmartICMC foram cumpridos, e, portanto, a instanciação desta LPOS foi bem sucedida.

6.4 Objetivos da MTTPL

Conforme mencionado no Capítulo 4, a proposta da MTTPL deve obedecer aos seguintes objetivos:

- Permitir a instanciação de sistemas que auxiliem a visita a museus;
- Facilitar a incorporação de dispositivos IoT à modelagem atual;

O primeiro objetivo já é sabido que foi cumprido, pois o SmartICMC é uma instância da MTTPL que possui este objetivo como uma de suas características. As razões que fundamentam esta conclusão podem ser vistas na seção anterior deste capítulo.

Figura 43 – Bússola utilizada pelo Aplicativo Móvel para auxiliar a navegação.

Já o segundo objetivo pode ser verificado pelos *assets* gerados da MTTPL, e também do SmartICMC. De acordo com o modelo de classes da Figura 23, o *Backend* da MTTPL possui um *broker* mapeado para comunicar-se com o *Broker de Sensores*. Quando qualquer sensor IoT for mapeado em uma instância da MTTPL que não seja o *smartphone* dos usuários, basta instanciar este *broker* e implementar as funcionalidades necessárias no serviço correspondente. Assim, quaisquer adaptações necessárias para incluir novos dispositivos IoT requerem grandes mudanças neste serviço, porém apenas neste sistema. Isto significa que a incorporação de novos dispositivos IoT gera impacto significativo apenas em um dos serviços instanciados, e que as mudanças necessárias para distribuir os dados capturados destes dispositivos para os outros serviços são de menor porte.

Assim, a incorporação de novos dispositivos IoT em qualquer instância da MTTPL é facilitada graças à divisão do sistema em diferentes serviços, principalmente pela definição de um serviço responsável por lidar exclusivamente com esta integração. Desta forma, o segundo objetivo da MTTPL também foi alcançado, e, portanto, a proposta da MTTPL atende à demanda identificada no início deste trabalho.

6.5 Conclusões

Neste capítulo, foi apresentada uma breve discussão acerca dos artefatos obtidos ao final do processo de engenharia de aplicação do SmartICMC, analisando os resultados alcançados ao final do processo.

Como os objetivos traçados foram cumpridos, de acordo com as seções anteriores, a

modelagem da MTTPL foi capaz de instanciar um produto correto que atenda às demandas do museu onde opera. Desta forma, há indícios de que a MTTPL seja considerada aplicável para o domínio para o qual foi modelada, validando, assim, sua proposta. Outras formas de validação serão necessárias no futuro para comprovar sua adequação a outras aplicações do domínio.

Da conclusão acima, também é possível extrair, como resultado, que a MTTPL cumpriu seu papel como linha de produtos de *software*. Por ter um produto instanciado a partir desta, modelado como uma LPOS desde sua concepção, futuros produtos com *features* novas serão instanciados mais facilmente. Além disso, características podem ser trocadas com facilidade, uma vez que o sistema foi construído como uma LPS, acelerando, assim, a instanciação de produtos distintos. Pode-se concluir, portanto, que a MTTPL foi capaz de contribuir com um grau de reuso de *software* mais elevado, fornecendo, assim, respostas afirmativas para as questões de pesquisa que motivaram este trabalho.

CONCLUSÃO

Dentro do domínio dos sistemas de aprendizagem, foi feito um SM (FUZETO; BRAGA, 2016) visando delinear o estado-da-arte a respeito do uso de dispositivos IoT e sistemas *U-learning*. Dos resultados obtidos, notou-se um alto grau de retrabalho entre as soluções propostas, evidenciando a necessidade de propor e validar a aplicabilidade de técnicas de reúso de *software* para este domínio de sistemas.

Como o trabalho de Junior (2015) apresenta uma LPS para sistemas *M-learning*, optou-se por utilizar este trabalho como ponto de partida, juntamente com os resultados do SM, para elaborar uma nova proposta para solucionar tal problema. No entanto, a presença de dispositivos IoT levanta o problema que a incorporação destes dispositivos traz à instância de uma LPS, devido à forma de comunicação com cada dispositivo, que pode usar protocolos personalizados (ATZORI; IERA; MORABITO, 2010; AL-FUQAHA *et al.*, 2015). Assim, a solução encontrada que melhor se adequa a este problema é a orientação a serviços (ROSEN *et al.*, 2008).

Utilizando a abordagem de LPOS (LEE; KOTONYA, 2010), foi desenvolvida uma linha de produtos a partir do trabalho de Junior (2015) e dos requisitos levantados após consultar especialistas no domínio, bem como a partir do SM citado. Estes foram utilizados para modelar uma linha de produtos, destacando os pontos de comunalidade e variabilidade, mapeados em um modelo de *features*. Com este resultado em mãos, os grupos de características foram agrupados em serviços, que foram analisados segundo os requisitos da LPOS para definir as classes a serem instanciadas para cada serviço da linha de produtos.

7.1 Contribuições

Este trabalho apresenta a MTTPL, uma LPOS proposta para solucionar o problema de reúso de *software* no domínio dos sistemas de auxílio à visitaço de museus. Alm de fornecer *assets* que permitem gerar novas instncias desta linha de produtos de acordo com o mapeamento

de características feito, a validação da MTTPL como solução aceitável para o problema apontado evidencia a possibilidade de aplicar a técnica de LPOS para o domínio de aplicação explorado por este trabalho.

Uma prova de conceito também foi realizada neste trabalho, o que gerou o sistema referido por SmartICMC. Este opera no Museu da Fauna e da Flora do ICMC, auxiliando os visitantes a encontrar diferentes exemplares de diferentes espécies de árvores plantadas pelo Instituto, além de apresentar cartões informativos com descrições de cada espécime. Uma verificação superficial do SmartICMC, baseada em inspeções de artefatos e código-fonte, resultou na inferência de que o sistema gerado atende às demandas estabelecidas para o mesmo, e que os aspectos de qualidade oferecidos pela MTTPL persistiram.

O resultado obtido com o SmartICMC é um sistema de auxílio à visitação de museus, mas que pode ser reaproveitado em qualquer contexto de aplicação semelhante a museus com itens em exibição estáticos. Parte dos códigos-fonte deste sistema, bem como os *links* de acesso aos repositórios, podem ser encontrados no [Capítulo 5](#).

7.2 Dificuldades e Limitações

A primeira dificuldade encontrada na condução deste trabalho foi em gerar os artefatos do SmartICMC adequadamente. Tomando o modelo de *features* como exemplo, que antes utilizava a linguagem proposta por [Kang et al. \(1990\)](#), gerava um modelo cujo tamanho era péssimo de ser visualizado neste documento. Para resolver este problema, adotou-se a representação em UML, proposta por [Gomaa \(2005\)](#). Assim, foi possível elaborar o modelo novamente, no padrão verticalizado que é introduzido no [Capítulo 4](#).

Durante o desenvolvimento do SmartICMC, um dos desafios identificados foi a restrição do aplicativo desenvolvido, relativo ao Sistema Operacional (SO) dos aparelhos. Originalmente, com o aplicativo sendo desenvolvido em Java através do *Android Studio*, o aplicativo seria exclusivo para dispositivos Android. Visando solucionar este problema, adotou-se o desenvolvimento baseado em JavaScript e React, o que permitiu o desenvolvimento rapidamente de um portal *web*, acessível de qualquer SO através do navegador. Além disso, o portal *web* desenvolvido se comporta como um aplicativo, provendo aos usuários uma experiência semelhante.

Outro desafio encontrado durante o desenvolvimento do portal *web* foi a leitura dos sensores dos *smartphones*, que era dependente de bibliotecas específicas para a geração de aplicativos instaláveis, indo contra a ideia inicial do aplicativo ter o aspecto de *portabilidade* fortalecido. O que permitiu contornar este problema foi a descoberta de que os navegadores, atualmente, estão provendo suporte à leitura de tais sensores. Assim, desde que o SmartICMC seja acessado através de um navegador com versão de atualização mais recente do que 2016, o usuário não deve encontrar nenhum problema na utilização do sistema.

Por fim, o último obstáculo encontrado foi a incerteza sobre quais espécimes de árvores do Museu da Fauna e da Flora capturar dados para inserir no SmartICMC. No entanto, o contato com a equipe de curadoria do museu resolveu este problema, uma vez que os envolvidos agiram rapidamente para providenciar os dados necessários, permitindo que o SmartICMC fosse alimentado com dados realísticos.

7.3 Trabalhos Futuros

Conforme mencionado no [Capítulo 5](#), a validação da MTTPL não foi feita neste trabalho devido à limitação no volume de tarefas a serem realizadas. Assim, a validação da linha de produtos proposta não é completa, restando validar cada uma das combinações de *features* instanciáveis.

A MTTPL, apesar de evidenciar a possibilidade de se aplicar a abordagem de LPOS para sistemas de aprendizagem, oferece garantias apenas para sistemas de auxílio à visita de museus. Isto gera um grau de incerteza acerca de todos os outros tipos de sistemas de aprendizagem, mesmo que a MTTPL se baseie em um SM de escopo aberto a todos os tipos. Portanto, se faz necessário investigar se os resultados obtidos pelo uso da abordagem de LPOS também são afirmativos, e se a aplicabilidade da MTTPL pode ser estendida a outros domínios de aplicação.

A MTTPL foi avaliada apenas utilizando a abordagem de *prova de conceito*. Como não foram encontradas soluções suficientemente semelhantes para conduzir uma análise comparativa, existem riscos em confiar plenamente na avaliação feita acerca da proposta deste trabalho. Sendo assim, se faz necessário levantar outras formas de avaliar a MTTPL e submetê-la a tais processos avaliativos, para validar com maior profundidade suas qualidades e providenciar dados que possam comprovar ou refutar os resultados deste trabalho mais concretamente.

Ainda sobre a avaliação da MTTPL, a mesma não foi comparada com sistemas monolíticos, nem com sistemas elaborados utilizando outras técnicas de reúso de *software*. Portanto, também não se sabe se a abordagem de LPOS é, efetivamente, a mais apropriada para este domínio de aplicações. Isto levanta a necessidade de explorar tal problema, e levantar possíveis alternativas à MTTPL.

A avaliação feita sobre a MTTPL tem um viés puramente técnico, sob o olhar de um engenheiro de *software*. Isto significa que os mecanismos de aprendizagem implementados no SmartICMC não foram avaliados sob o olhar pedagógico, visando validar as técnicas utilizadas como pedagogicamente viáveis. Portanto, ainda é necessário avaliar se o uso da MTTPL para instanciar novos sistemas de auxílio à visita de museus gera algum impacto nos processos de aprendizado dos usuários, e se tais impactos são positivos ou negativos.

Por fim, conforme mencionado no [Capítulo 3](#), o SM utilizado como parte da revisão

bibliográfica deste trabalho está desatualizado, dado que a captação de estudos foi realizada em 2016. Assim, para sanar este problema, é necessário que ao menos uma atualização do SM seja feita.

REFERÊNCIAS

- ABIRACHED, W. **Business Rules Must Be Enforced by the API**. 2017. Disponível em: <<https://designingforscale.com/business-rules-must-be-enforced-by-the-api/>>. Citado na página 95.
- ABOWD, G. D.; DEY, A. K.; BROWN, P. J.; DAVIES, N.; SMITH, M.; STEGGLES, P. Towards a better understanding of context and context-awareness. In: SPRINGER. **International symposium on handheld and ubiquitous computing**. [S.l.], 1999. p. 304–307. Citado na página 33.
- AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE communications surveys & tutorials**, IEEE, v. 17, n. 4, p. 2347–2376, 2015. Citado nas páginas 28 e 111.
- ALGHAMDI, A.; SHETTY, S. Survey toward a smart campus using the internet of things. In: IEEE. **Future Internet of Things and Cloud (FiCloud), 2016 IEEE 4th International Conference on**. [S.l.], 2016. p. 235–239. Citado na página 63.
- ANGULAR. **Angular**. 2019. Disponível em: <<https://angular.io/>>. Citado na página 95.
- ATIF, Y.; MATHEW, S. A social web of things approach to a smart campus model. In: IEEE. **Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing**. [S.l.], 2013. p. 349–354. Citado na página 63.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado nas páginas 25, 28, 33 e 111.
- BACHMANN, F.; COHEN, S.; DONOHEE, P.; LITTLE, R.; NORTHROP, L. **A Framework for Software Product Line Practice, Version 5.0**. 2012. Disponível em: <<http://www.sei.cmu.edu/productlines/>>. Citado na página 41.
- BADR, Y.; MAAMAR, Z. Autonomic service-oriented grid to enhance e-learning experiences. In: IEEE. **Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on**. [S.l.], 2009. p. 649–654. Citado na página 59.
- BAELDUNG. **The DAO Pattern in Java**. 2018. Disponível em: <<https://www.baeldung.com/java-dao-pattern>>. Citado na página 80.
- BOYINBODE, O.; BAGULA, A. An adaptive and personalized ubiquitous learning middleware support for handicapped learners. In: IEEE. **Information Technology: New Generations (ITNG), 2011 Eighth International Conference on**. [S.l.], 2011. p. 632–637. Citado na página 61.
- CABALLÉ, S. Towards a multi-modal emotion-awareness e-learning system. In: IEEE. **Intelligent Networking and Collaborative Systems (INCOS), 2015 International Conference on**. [S.l.], 2015. p. 280–287. Citado na página 65.

- CHANG, C.-W.; LIN, P.; TSENG, C.-W.; KONG, Y.-K.; LIEN, W.-C.; WU, M.-C.; WU, C.-Y. Poster: Design and implementation of mobile e-learning platform for medical training. In: ACM. **Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing**. [S.l.], 2015. p. 385–386. Citado na página 65.
- CHANG, W.-J.; YEH, Z.-M.; CHENG, K.-J. Developing a mobile learning system to community-based learning for rural elementary school students. In: IEEE. **Computer Symposium (ICS), 2010 International**. [S.l.], 2010. p. 451–456. Citado na página 26.
- CHAO, K.-M.; JAMES, A. E.; NANOS, A. G.; CHEN, J.-H.; STAN, S.-D.; MUNTEAN, I.; FIGLIOLINI, G.; REA, P.; BOUZGARROU, C. B.; VITLIEMOV, P. *et al.* Cloud e-learning for mechatronics: Clem. **Future Generation Computer Systems**, Elsevier, v. 48, p. 46–59, 2015. Citado na página 62.
- CHEN, T.-S.; CHANG, C.-S.; LIN, J.-S.; YU, H.-L. Context-aware writing in ubiquitous learning environments. **Research and Practice in Technology Enhanced Learning**, World Scientific, v. 4, n. 01, p. 61–82, 2009. Citado na página 62.
- CHENG, Z.; SUN, S.; KANSEN, M.; HUANG, T.; HE, A. A personalized ubiquitous education support environment by comparing learning instructional requirement with learner's behavior. In: IEEE. **Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on**. [S.l.], 2005. v. 2, p. 567–573. Citado na página 26.
- CHU, H.-C.; HWANG, G.-J. A location-aware mobile learning system to provide field learning guidance for natural science courses. In: IEEE. **Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on**. [S.l.], 2010. v. 3, p. 291–294. Citado na página 26.
- CHUKMOL, U.; BENHARKAT, A.-N.; AMGHAR, Y. Towards a user-oriented framework for web service discovery, reuse and evolution. In: IEEE. **Signal-Image Technology & Internet-Based Systems (SITIS), 2009 Fifth International Conference on**. [S.l.], 2009. p. 415–422. Citado na página 38.
- CLEMENTS, P.; NORTHROP, L. **Software product lines**. [S.l.]: Addison-Wesley, 2002. Citado nas páginas 15, 26 e 42.
- COCCOLI, M.; GUERCIO, A.; MARESCA, P.; STANGANELLI, L. Smarter universities: A vision for the fast changing digital era. **Journal of Visual Languages & Computing**, Elsevier, v. 25, n. 6, p. 1003–1011, 2014. Citado na página 63.
- COPLIEN, J.; HOFFMAN, D.; WEISS, D. Commonality and variability in software engineering. **IEEE software**, IEEE, v. 15, n. 6, p. 37–45, 1998. Citado na página 41.
- DONEGAN, P. M. **Geração de famílias de produtos de software com arquitetura baseada em componentes**. Tese (Doutorado) — Universidade de São Paulo, 2008. Citado nas páginas 15 e 44.
- EL-BISHOUTY, M. M.; OGATA, H.; YANO, Y. A model of personalized collaborative computer support ubiquitous learning environment. In: IEEE. **2008 Eighth IEEE International Conference on Advanced Learning Technologies**. [S.l.], 2008. p. 97–101. Citado na página 63.
- FACEBOOK. **React.js**. 2019. Disponível em: <<https://reactjs.org/>>. Citado na página 95.

FAYAD, M.; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, ACM, v. 40, n. 10, p. 32–38, 1997. Citado na página 57.

FIGLIORE, A.; MAINETTI, L.; PATRONO, L.; VERGALLO, R. An epc-based middleware enabling reusable and flexible mixed reality educational experiences. In: IEEE. **Software, Telecommunications and Computer Networks (SoftCOM), 2013 21st International Conference on**. [S.l.], 2013. p. 1–6. Citado na página 61.

FUZETO, R.; BRAGA, R. Um mapeamento sistemático em progresso sobre internet das coisas e educação à distância. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l.: s.n.], 2016. v. 5, n. 1, p. 1334. Citado nas páginas 28, 49 e 111.

GEEKERY, D. **jOOQ: The easiest way to write SQL in Java**. 2009. Disponível em: <<https://www.jooq.org/>>. Citado na página 94.

GOMAA, H. Designing software product lines with uml. In: **SEW Tutorial Notes**. [S.l.: s.n.], 2005. p. 160–216. Citado nas páginas 15, 26, 40, 41, 42, 43, 44, 45, 65, 67, 89 e 112.

GOMAA, H.; FARRUKH, G. Composition of software architectures from reusable architecture patterns. In: ACM. **Proceedings of the third international workshop on Software architecture**. [S.l.], 1998. p. 45–48. Citado na página 41.

GONZÁLEZ, M. Á.; GONZÁLEZ, M. Á.; LLAMAS, C.; MARTÍN, M. E.; VEGAS, J.; MARTÍNEZ, Ó.; HERNÁNDEZ, C.; HERGUEDAS, M. Mobile phones for teaching physics: using applications and sensors. In: ACM. **Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality**. [S.l.], 2014. p. 349–355. Citado na página 26.

GONZÁLEZ, M. Á.; SILVA, J. B. da; CAÑEDO, J. C.; HUETE, F.; MARTÍNEZ, Ó.; ESTEBAN, D.; MANSO, J.; ROCHADEL, W.; GONZÁLEZ, M. Á. Doing physics experiments and learning with smartphones. In: ACM. **Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality**. [S.l.], 2015. p. 303–310. Citado na página 26.

HAKKE, R. R. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. **American journal of Physics**, AAPT, v. 66, n. 1, p. 64–74, 1998. Citado na página 31.

HE, L.; SAAD, A.; REED, J.; HANNIGAN, P.; STRAUSSER, E. Information technology education for k-12 students and teachers: from sensor network to comprehensive and customized web interaction. In: ACM. **Proceedings of the 9th ACM SIGITE conference on Information technology education**. [S.l.], 2008. p. 65–70. Citado na página 26.

HOU, B.; OGATA, H.; MIYATA, M.; LI, M.; YANO, Y. Development of web-based japanese mimicry and onomatopoeia learning assistant system with sensor network. In: IEEE. **Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), 2010 6th IEEE International Conference on**. [S.l.], 2010. p. 117–121. Citado na página 62.

HSU, J.-M.; LAI, Y.-S.; YU, P.-T. Using the rfids to construct the ubiquitous self-learning environment for understanding the plants in the schoolyard. In: IEEE. **Wireless, Mobile, and Ubiquitous Technology in Education, 2008. WMUTE 2008. Fifth IEEE International Conference on**. [S.l.], 2008. p. 210–212. Citado na página 65.

- HWANG, G.-J.; CHU, H.-C.; LIN, Y.-S.; TSAI, C.-C. A knowledge acquisition approach to developing mindtools for organizing and sharing differentiating knowledge in a ubiquitous learning environment. **Computers & Education**, Elsevier, v. 57, n. 1, p. 1368–1377, 2011. Citado na página 63.
- HWANG, G.-J.; TSAI, C.-C.; YANG, S. J. Criteria, strategies and research issues of context-aware ubiquitous learning. **Journal of Educational Technology & Society**, JSTOR, v. 11, n. 2, p. 81–91, 2008. Citado na página 28.
- HWANG, G.-J.; YANG, T.-C.; TSAI, C.-C.; YANG, S. J. A context-aware ubiquitous learning environment for conducting complex science experiments. **Computers & Education**, Elsevier, v. 53, n. 2, p. 402–413, 2009. Citado na página 33.
- JUNIOR, V. F. **Estudo e definição de uma linha de produtos de software para o desenvolvimento de aplicações educacionais móveis**. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado nas páginas 27, 64, 67 e 111.
- KAISER, R.; OERTEL, K. Emotions in hci: an affective e-learning system. In: AUSTRALIAN COMPUTER SOCIETY, INC. **Proceedings of the HCSNet workshop on Use of vision in human-computer interaction-Volume 56**. [S.l.], 2006. p. 105–106. Citado na página 26.
- KANG, K. C.; COHEN, S. G.; HESS, J. A.; NOVAK, W. E.; PETERSON, A. S. **Feature-oriented domain analysis (FODA) feasibility study**. [S.l.], 1990. Citado nas páginas 42 e 112.
- KEARNEY, M.; SCHUCK, S.; BURDEN, K.; AUBUSSON, P. Viewing mobile learning from a pedagogical perspective. **Research in learning technology**, v. 20, 2012. Citado na página 32.
- KIM, S.; SONG, S.-M.; YOON, Y.-I. Smart learning services based on smart cloud computing. **Sensors**, Molecular Diversity Preservation International, v. 11, n. 8, p. 7835–7850, 2011. Citado na página 60.
- KITCHENHAM, B.; CHARTERS, S.; BUDGEN, D.; BRERETON, P.; TURNER, M.; LINKMAN, S.; JØRGENSEN, M.; MENDES, E.; VISAGGIO, G. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.], 2007. Citado na página 49.
- LARMAN, C. **Utilizando UML e padrões**. [S.l.]: Bookman Editora, 2002. Citado na página 69.
- LEE, J.; KOTONYA, G. Combining service-orientation with product line engineering. **IEEE software**, IEEE, v. 27, n. 3, p. 35–41, 2010. Citado nas páginas 27, 46, 65, 74 e 111.
- LEE, J.; MUTHIG, D.; NAAB, M. An approach for developing service oriented product lines. In: IEEE. **Software Product Line Conference, 2008. SPLC'08. 12th International**. [S.l.], 2008. p. 275–284. Citado nas páginas 45 e 67.
- LIU, T.-Y.; CHU, Y.-L.; TAN, T.-H.; CHANG, C.-C. Rfid-based ubiquitous learning environment for outdoor learning. In: IEEE. **Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on**. [S.l.], 2007. p. 675–677. Citado na página 65.
- MAURER, H. Necessary aspects of quality in elearning systems. In: **Quality education@ a distance**. [S.l.]: Springer, 2003. p. 29–37. Citado na página 32.

MCPARLAND, M.; NOBLE, L. M.; LIVINGSTON, G. The effectiveness of problem-based learning compared to traditional teaching in undergraduate psychiatry. **Medical education**, Wiley Online Library, v. 38, n. 8, p. 859–867, 2004. Citado na página 31.

MENDONCA, F. **Front End Architecture — Making rebuild from scratch not so painful**. 2018. Disponível em: <<https://medium.com/car2godevs/front-end-architecture-making-rebuild-from-scratch-not-so-painful-7b2232dc1666>>. Citado na página 95.

MOOGK, D. R. Minimum viable product and the importance of experimentation in technology startups. **Technology Innovation Management Review**, v. 2, n. 3, 2012. Citado na página 89.

MOZILLA. **Browser support for JavaScript APIs**. 2019. Disponível em: <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Browser_support_for_JavaScript_APIs>. Citado na página 95.

NETLIFY. **Netlify: All-in-one platform for automating modern web projects**. 2019. Disponível em: <<https://www.netlify.com/>>. Citado na página 96.

OGATA, H. Computer supported ubiquitous learning: Augmenting learning experiences in the real world. In: IEEE. **Wireless, Mobile, and Ubiquitous Technology in Education, 2008. WMUTE 2008. Fifth IEEE International Conference on**. [S.l.], 2008. p. 3–10. Citado na página 61.

_____. Augmenting learning-experiences in the real world with digital technologies. In: IEEE. **Recent Trends in Information Technology (ICRTIT), 2013 International Conference on**. [S.l.], 2013. p. 237–243. Citado na página 26.

OGATA, H.; YANO, Y. Context-aware support for computer-supported ubiquitous learning. In: IEEE. **The 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, 2004. Proceedings**. [S.l.], 2004. p. 27–34. Citado na página 63.

ORACLE. **O que é o Java?** 2019. Disponível em: <https://www.java.com/pt_BR/about/whatis_java.jsp?bucket_value=desktop-chrome72-windows10-64bit&in_query=no>. Citado na página 93.

_____. **Why MySQL?** 2019. Disponível em: <<https://www.mysql.com/why-mysql/>>. Citado na página 94.

PALETTA, L.; DUNWELL, I.; GAVED, M.; BOBETH, J.; EFREMIDIS, S.; LULEY, P.; KUKULSKA-HULME, A.; FREITAS, S. de; LAMERAS, P.; DEUTSCH, S. Advances in maseltov—serious games in a mobile ecology of services for social inclusion and empowerment of recent immigrants. In: **Advances in Computer Entertainment**. [S.l.]: Springer, 2013. p. 440–455. Citado na página 61.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMAN, F. Restful web services vs. big web services: making the right architectural decision. In: ACM. **Proceedings of the 17th international conference on World Wide Web**. [S.l.], 2008. p. 805–814. Citado nas páginas 39 e 40.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08**. Swinton, UK, UK: British Computer Society, 2008. p. 68–77. Citado na página 49.

- PIVOTAL. **Spring Boot**. 2019. Disponível em: <<http://spring.io/projects/spring-boot>>. Citado na página 94.
- PRESSER, M.; KRICO, S.; KOWATSCH, T.; MAASS, W.; LANGE, S.; CARREZ, F.; HUNT, B.; EGAN, R.; HÖLLER, J.; BASSI, A. *et al.* Inspiring the internet of things: The internet of things comic book. Alexandra Institute, 2011. Citado nas páginas 33 e 34.
- RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: "O'Reilly Media, Inc.", 2008. Citado nas páginas 38 e 39.
- ROSEN, M.; LUBLINSKY, B.; SMITH, K. T.; BALCER, M. J. **Applied SOA: service-oriented architecture and design strategies**. 1. ed. [S.l.]: Wiley Publishing Inc., 2008. 662 p. Citado nas páginas 15, 26, 27, 35, 36, 37, 65 e 111.
- SALESFORCE. **Heroku: Cloud Application Platform**. 2019. Disponível em: <<https://www.heroku.com/>>. Citado na página 96.
- SHEN, L.; WANG, M.; SHEN, R. Affective e-learning: Using “emotional” data to improve learning in pervasive learning environment. **Journal of Educational Technology & Society**, JSTOR, v. 12, n. 2, p. 176–189, 2009. Citado na página 32.
- SILVA, R.; ANTONOVA, N.; SILVA, J.; MENDES, A.; MARCELINO, M. Wireless sensor networks to support elementary school learning activities. In: ACM. **Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing**. [S.l.], 2009. p. 71. Citado na página 26.
- SUN. **Core J2EE Patterns - Transfer Object**. 2002. Disponível em: <<https://www.oracle.com/technetwork/java/transferobject-139757.html>>. Citado na página 80.
- VERBORGH, R.; STEINER, T.; DEURSEN, D. V.; WALLE, R. V. de; GABARRO, J. Efficient runtime service discovery and consumption with hyperlinked restdesc. In: **The 7th International Conference on Next Generation Web Services Practices (NWeSP 2011)**. Salamanca, Spain: [s.n.], 2011. Disponível em: <<http://www.lsi.upc.edu/~tsteiner/papers/2011/efficient-runtime-service-discovery-nwesp2011.pdf>>. Citado nas páginas 26 e 37.
- VILLANUEVA, V. J. A.; MARQUEZ, F. J. D.; SOLARTE, A. Z. M.; DÁVALOS, A. G. Architecture for integrating real objects with virtual academic communities. In: IEEE. **e-Learning (econf), 2015 Fifth International Conference on**. [S.l.], 2015. p. 385–391. Citado nas páginas 60 e 62.
- VUE.JS. **Introduction to Vue.js**. 2019. Disponível em: <<https://vuejs.org/v2/guide/>>. Citado na página 95.
- WU, T.-T.; HUANG, Y.-M.; CHAO, H.-C.; PARK, J. H. Personalized english reading sequencing based on learning portfolio analysis. **Information Sciences**, Elsevier, v. 257, p. 248–263, 2014. Citado na página 62.
- XUE, R.; WANG, L.; CHEN, J. Using the iot to construct ubiquitous learning environment. In: IEEE. **Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on**. [S.l.], 2011. p. 7878–7880. Citado na página 26.
- YANG, H. Y.; TEMPERO, E.; MELTON, H. An empirical study into use of dependency injection in java. In: IEEE. **19th Australian Conference on Software Engineering (aswec 2008)**. [S.l.], 2008. p. 239–247. Citado na página 94.

ZAPPATORE, M.; LONGO, A.; BOCHICCHIO, M. A.; ZAPPATORE, D.; MORRONE, A. A.; MITRI, G. D. A crowdsensing approach for mobile learning in acoustics and noise monitoring. In: ACM. **Proceedings of the 31st Annual ACM Symposium on Applied Computing**. [S.l.], 2016. p. 219–224. Citado na página [26](#).

GLOSSÁRIO

Backend: Parte de um sistema de *software* referente ao que está presente próximo ao servidor. Em geral, são sistemas responsáveis por manipular os dados do sistema, aplicando regras de negócio.

Cache: Espaço reservado no disco rígido de um usuário para armazenar algumas informações de páginas *web*. Remove a necessidade de recuperar todas as informações de uma página, reduzindo a largura de banda necessária para carregá-la.

Caching: Utilização do *cache* de uma página *web*.

Frontend: Parte de um sistema de *software* referente ao que o usuário vê e interage. Exemplos: aplicativo instalado no celular, página *web*, *software* instalado no computador.

Renderizar: Ato de inserir as informações de um componente do *frontend* no código HTML da página, a fim de ser exibido.

Script: Arquivo de código-fonte com um objetivo extremamente específico.

ESTUDOS SELECIONADOS PELO SM

Devido ao grande volume de estudos e dados coletados, o espaço tomado pelo título desta página também é necessário para exibir a tabela. Por isso, a mesma foi incluída na página a seguir.

Title	Author	Year	Country	Type of study	Methods used	Type of contribution	Science field where it is applied	Scholarship level of students	Type of learning environment	IoT technology types used	Does the author recognize the role of expertise in the field?	Educational resources used
Ubiquitous Learning Analytics in the Context of Real-world Language Learning	Mout, K; Ogata, H; Usaki, N.	2015	Japan	Theoretical	System development	Tool	Language	Not specified/Does not apply	U-learning	WSNs, Cloud-centered systems	No	Learning log
Doing Physics Experiments and Learning with Smartphones	Gonzalez, M. Silva, J.B., Canedo, J.C., Huete, F., Martinez-Gonzalez, S. W	2015	Spain	Experimental	System development, Experimental execution	Tool	Physics	Not specified/Does not apply	None of the above	WSNs	No	Chart
Responsive Facilitation of Experiential Learning Through Access to Attentional State	Gonzalez, M., Silva, J., Canedo, J., Huete, F., Martinez-Gonzalez, S. W	2015	Spain	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	M-learning	WSNs	No	Picture
Mobile Phones for Teaching Physics: Using Applications and Sensors	Gonzalez, M., Silva, J., Canedo, J., Huete, F., Martinez-Gonzalez, S. W	2014	Spain	Experimental	System development, Experimental execution	Tool	Physics	Not specified/Does not apply	M-learning	WSNs	No	Smartphone app
U-Like a Ubiquitous Computing based architecture to locate works using Wireless Sensor Network and Radio-Frequency Identification	Sarmiento, W.F.F.; Paula, P.S.; Pequeno Filho, P.T.C.; Pali, G.H. S.H.	2012	Brazil	Experimental	System development, Experimental execution	Tool	Chemistry	Bachelor or equivalent	U-learning	RFID tags, WSNs	No	Learning-oriented environment
Cloud Computing and Mapping for Reliability and Scalability of Ubiquitous Learning Systems	Shin, R., Anwar, N., Shin, S., Mendes, A., Marcolino, M.	2011	USA	Experimental	Experimental execution	Architecture	None	Not specified/Does not apply	U-learning	Cloud-centered systems	No	Learning-oriented environment
Wireless Sensor Networks to Support elementary school learning activities	Shin, R., Anwar, N., Shin, S., Mendes, A., Marcolino, M.	2009	Portugal	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	M-learning	WSNs	No	Learning-oriented environment
Distributed Weather Net: Wireless Sensor Network Supported Inquiry-Based Learning	Chang, B., Wang, H.Y., Chen, C.S., Liang, J., Lamp, F., Kopf, S., Effenberg, W.	2009	Taiwan	Experimental	System development, Experimental execution	Method	Weather	Lower secondary	E-learning	WSNs	No	chart
Automatic Lecture Recording	Lamp, F., Kopf, S., Effenberg, W.	2008	Germany	Experimental	System development	Method	None	Not specified/Does not apply	None of the above	WSNs	No	Lecture, Video
Information Technology Education for K-12 Students and Teachers: From Sensor Network to Comprehensive and Customized Web Interaction	Hu, J., Saeed, A., Reed, J., Hengpan, P., Strauser, E.	2008	USA	Theoretical	System development, Experimental execution	Tool	IT	Upper secondary	E-learning	WSNs	No	Picture, Book/Text, Lecture, Video, Learning-oriented environment
Environments in HCI - An Effective E-Learning System	Kaiser, R., Omer, K.	2006	Germany	Experimental	Experimental execution	Survey	None	Not specified/Does not apply	E-learning	WSNs	No	Picture, Book/Text
Computer-Supported Ubiquitous Learning: Augmenting Learning Experiences in the Real World	Ogata, H., Yano, Y.	2004	Japan	Experimental	System development	Tool	Language	Post-secondary non-tertiary	U-learning	PDA	No	Book/Text, Helper
A Personalized Ubiquitous Learning Support Environment by Comparing Learning Instructional Requirement with Learner's Behavior	Cheng, Z., Sun, S., Karsten, M., Huang, T., He, A.	2005	Japan	Theoretical	System development	Framework	None	Not specified/Does not apply	U-learning	RFID tags, WSNs	No	Activity record
Ubiquitous Computing Technologies for Ubiquitous Learning	Sakuma, K., Kashiwaba, N.	2005	Japan	Experimental	Case study	Survey	None	Not specified/Does not apply	U-learning	None	No	Book/Text, Video
Specification of a Functional Architecture for E-Learning Supported by Wireless Technologies	Grew, P., Giudici, F., Pignati, E.	2006	Italy	Theoretical	System development	Architecture	None	Bachelor or equivalent	U-learning	RFID tags, WSNs, None	No	Learning-oriented environment
Criteria and Strategies of Ubiquitous Learning	Hwang, G.J.	2006	Taiwan	Theoretical	Case study	Survey	Biology	Lower secondary	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Learning-oriented environment
A Ubiquitous Learning System for Improving Quality of Learner's Behaviors Based on Shaping Principle	Sun, S., Zheng, Z., Jing, L., Huang, T., Hou, H.H.	2006	Japan	Experimental	System development, Experimental execution	Tool, Method	None	Not specified/Does not apply	U-learning	RFID tags, WSNs	No	Learning-oriented environment
Context-Aware Ubiquitous English Learning in a Campus Environment	Hsieh, H.C., Chen, C.M., Hong, C.M.	2007	Taiwan	Theoretical	System development	Tool, Architecture	Language	Not specified/Does not apply	U-learning	Mobile handheld devices	No	Book/Text, Learning-oriented environment
RFID-based Ubiquitous Learning Environment for Outdoor Learning	Liu, T.Y., Chu, Y.L., Tan, T.H., Chang, C.C.	2007	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Upper secondary	U-learning	RFID tags, Mobile handheld devices	No	None
Applying Wireless Sensor Networks to Context-Awareness in Ubiquitous Learning	Wang, M., Li, L., Zhan, P., Xu, Y.	2007	China	Theoretical	Case study	Survey	None	Not specified/Does not apply	U-learning	WSNs	No	None
Computer-Supported Ubiquitous Learning: Augmenting Learning Experiences in the Real World	Ogata, H.	2008	Japan	Experimental	System development, Experimental execution	Tool	Language	Not specified/Does not apply	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Video, Learning-oriented environment, Dictionary
Context-Aware Writing in Ubiquitous Learning Environments	Chen, T.S., Chang, C.S., Lin, J.S., Yu, H.L.	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Language	Lower secondary	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text, Video, Learning-oriented environment
Conducting Situated Learning in a Context-Aware Ubiquitous Learning Environment	Hou, B., Ogata, H., Yano, Y.	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Lower secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Learning-oriented environment
RFID-based Ubiquitous Learning Environment for School Students	Yin, J., Yang, X.	2008	China	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	RFID tags, Mobile handheld devices	No	None
Using the RFID to Construct the Ubiquitous Self-Learning Environment for Understanding the Plants in the Schoolyard	Hsu, J.M., Lai, Y.S., Yu, P.T.	2008	Taiwan	Theoretical	System development, Experimental execution	Tool	Ecology	U-learning	RFID tags, Mobile handheld devices	No	None	
A Model of Personalized Collaborative Computer-Supported Ubiquitous Learning Environment	El-Bishry, M.M., Ogata, H., Yano, Y.	2008	Japan	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	RFID tags	No	None
An Improved Design of the Ubiquitous Learning System Based on Sensor Networks	Dong, M., Zhang, G., Qiu, K., Guo, S., Guo, M.	2008	China	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	RFID tags, WSNs	No	None
Handheld Augmented Reality Cloud-based Immersive Ubiquitous Learning System	Liu, T.Y., Chang, C.C.	2008	Taiwan	Theoretical	System development, Experimental execution	Tool, Method	Language	Not specified/Does not apply	U-learning	RFID tags, Mobile handheld devices	No	None
Fundamental Framework of Supporting M-learning in RFID System	Lin, C.C., Shih, D.H.	2008	Taiwan	Theoretical	System development	Tool	None	Not specified/Does not apply	M-learning	RFID tags, Mobile handheld devices	No	None
Middleware for the development of context-aware applications inside m-learning: Connecting e-learning to the mobile world	Martin, S., Gil, R., Cristobal, E.S., Diaz, G., Castro, M., Perez, Chong-Puy, Y.	2009	Spain	Theoretical	System development	Middleware	None	Not specified/Does not apply	U-learning	RFID tags	No	None
Personalized guidance and Ubiquitous learning in Intelligent Learning with Multi-Agent	Chen, H.C., Hwang, G.J.	2010	Taiwan	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	RFID tags, Mobile handheld devices	No	Book/Text
A location-aware mobile learning system to provide field learning guidance for natural science classes	Chen, H.C., Hwang, G.J.	2010	Taiwan	Experimental	System development, Experimental execution	Tool, Technique	Mathematics	Lower secondary	E-learning	RFID tags, WSNs, Mobile handheld devices	No	Learning-oriented environment, Learning objects
Developing Ubiquitous Learning System with Robots for Children's Learning	Chang, B., Chen, C.W.	2010	Taiwan	Experimental	System development, Experimental execution	Technique	Mathematics	Bachelor or equivalent	E-learning	WSNs	No	Picture, Book/Text, Learning-oriented environment
Supporting Ubiquitous Learning in a Context-Aware Ubiquitous Learning Environment	Tsang, J.C.R., Hsu, C.H., Hwang, G.J.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	None
An Enhanced Context-Sensitive Initialization-Based m-learning Architecture	Meene, D., Garache, I., O'Driscoll, M., Stojanov, S., Vahonov, Chang, W.J., Yeh, Z.M., Cheng, K.J.	2010	Ireland	Experimental	Case study	Survey	None	Not specified/Does not apply	M-learning	Mobile handheld devices	No	None
Developing a Mobile Learning System to Community-based Learning for Rural Elementary School Students	Boydinbo, O., Bagdas, A.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	Agronomy	Upper secondary	M-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Video, Audio
An Adaptive and Personalized Ubiquitous Learning Middleware Support for Handcapped Learners	Patelides, R.D., Joon, S.H., Kim, T.H.	2011	South Africa	Theoretical	System development	Middleware	None	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	None
U-learning Community: An Intuitive Social Learning Model Based on Wireless Sensor Networks	Wong, S.F., Ho, Y.M., Yang, Z., Kwok, C.T.	2011	China	Theoretical	System development, Experimental execution	Tool	None	Not specified/Does not apply	U-learning	RFID tags	No	None
RFID-Aided Manufacturing Training System and Localization	Kuo, D.T., Chang, C.C.	2010	China	Experimental	System development, Experimental execution	Tool	Language	Bachelor or equivalent	U-learning	RFID tags, Cloud-centered systems, mobile handheld devices	No	Book/Text, Interactive activities
Development of Context Awareness Learning System for Elementary Chinese Language Learning	Johannes, D., Stronig, M., Hansen, R.B.A., Christensen, J., Sangheeth, K., Pradeep, P., Divya, P., Rekha, P., Arya, D.R.D.	2015	Norway	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	E-learning	WSNs	No	Hand-on-activities
A Remote Code Editing Framework for ARMIST Remote Triggered WSN Laboratory	Chang, Y.H., Lin B.S.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	History	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text
An Inquired Bayesian Network to construct the adaptive Personalized E-Learning Platform	Yao, C.B.	2010	Taiwan	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text
Augmenting Learning Experiences in the Real World with Digital Media	Ogata, H.	2010	Taiwan	Theoretical	System development, Experimental execution	Tool	Language	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text, Learning logs
Development of a Web 2.0-based Ubiquitous Learning Platform for Schooled Plant Identification	Ogata, H.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Lower secondary	U-learning	QR code, WSNs	No	Picture, Book/Text
Development of Web-based Japanese Memory and Chronotopical Learning Assistant System with Sensor Network	Hsu, B., Ogata, H., Miyata, M., Li, M., Yano, Y.	2010	Japan	Experimental	System development, Experimental execution	Tool	Linguistics	Not specified/Does not apply	U-learning	RFID tags, WSNs, Cloud-centered systems, Mobile handheld devices	No	Book/Text, Dictionary
Matrix Analysis of Collaborative Activities Based on a Meeting gateway Sensors for Real Classrooms	Chen, H.C., Hwang, G.J.	2010	Taiwan	Experimental	Case study, System development	Framework	IT	Upper secondary	M-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Video
Implementation of Ubiquitous Learning System Using Sensor Technologies	Mandula, K. Neda, S.R., Jan, D.K., Kambar, R.	2010	India	Experimental	System development, Experimental execution	Tool	None	Bachelor or equivalent	U-learning	RFID tags, WSNs, Mobile handheld devices	No	None
LORAMS: Sharing Learning Systems with Social and Ubiquitous Media	Ogata, H., Masumi, T., Hou, B., Li, M., El-Bishry, M., Yano, Y.	2010	Japan	Experimental	System development, Experimental execution	Tool	None	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Video
M2Learn: Towards a homogeneous vision of advanced mobile learning development	Martin, S., Perez, J., Castro, M.	2010	Spain	Theoretical	System development	Middleware	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	Learning-oriented environment
Recognizing the Emotion of Learners by Physiological Sensors to Improve English Learning Performance	Kuo, Y.C., Tseng, C.C.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	Linguistics	Bachelor or equivalent	U-learning	WSNs	No	None
Reliable Web-based CSCW Systems using Information Fusion of various multiple Biological Sensors	Sakurai, Y., Hashida, S., Tanaka, S., Ihara, H.	2008	Japan	Experimental	System development, Experimental execution	Tool	None	Bachelor or equivalent	U-learning	WSNs	No	None
A context-aware ubiquitous learning environment for conducting complex science experiments	Hwang, G.J., Yang, T.C., Tsai, C.C., Yang, S.J.H.	2009	Taiwan	Experimental	System development, Experimental execution	Tool	Chemistry	Bachelor or equivalent	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text, Learning-oriented environment
A knowledge acquisition approach to developing methods for learning and sharing offsetting knowledge in a ubiquitous learning environment	Hwang, G.J., Yang, T.C., Tsai, C.C., Yang, S.J.H.	2009	Taiwan	Experimental	System development, Experimental execution	Tool	Chemistry	Bachelor or equivalent	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text
A knowledge engineering approach to developing methods for learning and sharing offsetting knowledge in a ubiquitous learning environment	Hwang, G.J., Yang, T.C., Tsai, C.C., Yang, S.J.H.	2009	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Lower secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text
A pilot study on conducting mobile learning activities for clinical nursing courses based on the repository grid approach	Wu, P.H., Hwang, G.J., Tsai, C.C., Chen, Y.C., Huang, Y.M.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	Health	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Book/Text
A prototyped annotation application to conducting mobile learning activities for architecture design courses	Sung, H.Y., Hwang, G.J., Liu, S.Y., Chu, J.H.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	Architecture	Bachelor or equivalent	U-learning	Mobile handheld devices	No	Picture, Book/Text, Learning-oriented environment
A review of mobile pervasive learning applications and issues	Shin, L., Shamsuddin, S., Samal, M.H.	2010	India	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	None
Ubiquitous English vocabulary learning system: Evidence of affective/attitudinal vs. usefulness/ease-of-use	Huang, Y.M., Huang, Y.M., Huang, S.H., Lin, Y.T.	2012	Taiwan	Experimental	System development, Experimental execution	Framework	Linguistics	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Book/Text, Video
Blended learning system for efficient professional training	Rynda, A., Parada, X.G., Garcia, R., Diaz, G., Martinez, D.I.	2010	Spain	Experimental	System development, Experimental execution	Tool	None	Not specified/Does not apply	U-learning	WSNs, Mobile handheld devices	No	Book/Text
Current status, opportunities and challenges in education	Hsu, H.K., Lee, S.W., Chang, H.Y., Liang, J.C.	2013	Taiwan	Experimental	Case study	Survey	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	Augmented reality
Here and now mobile learning: An experimental study on the use of mobile technology	Martin, F., Erdinger, J.	2013	USA	Experimental	Experimental execution	Experimental analysis	History	Bachelor or equivalent	M-learning	Mobile handheld devices	No	Picture, Book/Text, Audio
Learning in a u-Museum: Developing a context-aware ubiquitous learning environment	Chen, C.C., Huang, T.C.	2012	Taiwan	Experimental	System development, Experimental execution	Tool	History	Upper secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Video
Personalized English reading sequencing based on learning portfolio analysis	Wu, T.T., Huang, Y.M., Chao, H.C., Park, J.H.	2014	Taiwan	Experimental	System development, Experimental execution	Framework	Linguistics	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Book/Text
Advances in MUSETOV - Serious Games in Mobile Ecology of Services for Social Inclusion and Empowerment of Recent Immigrants	Patelides, R.D., Joon, S.H., Kim, T.H.	2015	Taiwan	Experimental	System development, Experimental execution	Tool	Linguistics	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Learning-oriented environment, Serious games
A hybrid intelligent design approach to affective-sensitive learning	Kuo, Y.C., Lin, B.S., Liu, Y.L., Zhang, K., Kretzschmar, J.	2016	China	Experimental	System development, Experimental execution	Tool, Framework	Linguistics	Bachelor or equivalent	E-learning	WSNs	No	Book/Text, Learning-oriented environment
Remote Web-based Control Laboratory for Mobile Devices based on ELAS, Raspberry Pi and Node.js	Bernardo-Ortega, J., Baeza-Pardo, E., Lopez-Orozco, J.J.	2015	Spain	Theoretical	System development	Tool	Physics	Not specified/Does not apply	M-learning	WSNs, Mobile handheld devices	No	Learning-oriented environment
Signify: An alternative learning environment for learning and sharing offsetting knowledge in a ubiquitous learning environment	Chen, H.C., Hwang, G.J.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	Linguistics	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Dictionary
Technological support for the enactment of collaborative scripted learning activities across multiple spatial locations	de la Fuente-Valentin, L., Pérez-Sanz, M., Hernández, Wong, L.H., Looi, C.K.	2010	Spain	Experimental	System development, Experimental execution	Tool	IT	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text
What seems do we remove in mobile-assisted learning research? A critical review of the literature	Wong, L.H., Looi, C.K.	2011	Singapore	Experimental	Case study	Survey	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	none
Using ubiquitous games in an English listening and speaking course: Impact on learning outcomes and motivation	Liu, T.Y., Chu, Y.L.	2010	Taiwan	Experimental	System development, Experimental execution	Tool	Linguistics	Post-secondary non-tertiary	U-learning	Mobile handheld devices	No	Book/Text, Audio
Ubiquitous learning website: Scaffold learners by mobile devices with information-aware techniques	Chen, C.C., Chang, C.K., Wang, C.Y.	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Bachelor or equivalent	U-learning	Mobile handheld devices	No	None	
Organization of e-learning in a pervasive campus - a case study	Tawangarini, D., Nottling, K., Lucio, A.	2006	Germany	Experimental	Case study	Survey	None	Not specified/Does not apply	M-learning	Mobile handheld devices	No	none
A model for learning objects adaptation in light of mobile and context-aware computing	Abeon, M., da Costa, C.A., Barbosa, J.L.V., Rijo, B.J., Right, Tatsuura, D., Kato, M., Tamar, S., Specht, M.	2015	Brazil	Experimental	System development, Experimental execution	Model	None	Bachelor or equivalent	U-learning	Mobile handheld devices	No	Picture, Book/Text, Video, Question, Audio
Mobile authoring of open educational resources for authentic learning scenarios	Torreiro, R.A.W., Graf, S.	2016	Netherlands	Experimental	Case study, System development, Experimental execution	Tool	Systematic review/mapping	Bachelor or equivalent	E-learning	Mobile handheld devices	No	Picture, Book/Text, Video, Question, Audio
Considering learning styles and context-awareness for mobile adaptive learning	Jeong, H.Y., Hong, B.H.	2013	Finland	Experimental	System development, Experimental execution	Tool	Physics	Post-secondary non-tertiary	U-learning	WSNs, Mobile handheld devices	No	Picture, Book/Text, Lecture, Video, Learning-oriented environment, Audio
A practical use of learning system using user preference in ubiquitous computing environment	Zimmermann, A., Specht, M., Lorenz, A.	2013	South Korea	Experimental	System development, Experimental execution	Tool	Linguistics	Bachelor or equivalent	U-learning	Mobile handheld devices	No	Book/Text
Personalization and Content Management	Zyglid, J.M., Warner, Z.	2005	Germany	Theoretical	System development	Framework	None	Not specified/Does not apply	U-learning	WSNs, Mobile handheld devices	No	Picture, Book/Text
Mobile apps for science learning: Review of research	Shen, C.W., Wu, Y.C., Li, L.	2016	USA	Experimental	Experimental execution	Framework	IT	Bachelor or equivalent	M-learning	Mobile handheld devices	No	Learning-oriented environment
Developing a NFC-equipped smart classroom: Effects on attitudes toward computer science	Wang, S.J., Wu, Y.C.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	SCORM
Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system	Zyglid, J.M., Warner, Z.	2005	Germany	Theoretical	System development	Framework	None	Not specified/Does not apply	U-learning	WSNs, Mobile handheld devices	No	Learning-oriented environment
A Comprehensive Approach for Mobile Learning in Automatic Context-Aware Learning	Gargallo, L., Murillo, C.H.	2010	Spain	Experimental	System development, Experimental execution	Tool	Physics	Bachelor or equivalent	M-learning	WSNs	No	Games
A Novel Sensor-Based Methodology for Learner's Motivation Analysis in Game-Based Learning	Cheng, Z., Han, Q., Sun, S., Karsten, M., Hosokawa, T.	2006	Japan	Experimental	Theory deep analysis	Theory review	None	Not specified/Does not apply	U-learning	RFID tags, WSNs	No	none
A proposal on a Learner's Context-aware Personalized Education Support Method based on Principles of Behavior Science	Ye, X., Jing, L., Karsten, M., Wang, Y., Qiu, K.A.	2010	Japan	Experimental	System development, Experimental execution	Method	Health	Not specified/Does not apply	U-learning	RFID tags, Mobile handheld devices	No	none
Support mobile with changeable learning strategies based on mutual adaptation between an ubiquitous pet and a learner	Choi, S.J., Kang, J.M.B.	2012	South Korea	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	none
An Adaptive System Supporting Collaborative Learning Based on a Location-Based Social Network and Semantic Understanding	Badt, Y., Maam, Z.B.	2009	France	Theoretical	System development	Architecture	None	Not specified/Does not apply	E-learning	WSNs	No	none
Optimized Service-Oriented Grids for E-Learning Experiences	Fan, S., Fan, J., Zhang, Y., He, Z.	2008	China	Theoretical	System development	Method	None	Not specified/Does not apply	M-learning	Mobile handheld devices	No	none
An Automated System for Mobile Learning on IP-Based Learning Using SIP	Lee, B., Lee, R., Choi, M., Zhai, M.B.	2010	France	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	none
Analysis of an Agent-based Learning Environment	Ramos, D.V., Paimel, N., Ukash, M., Sumi, S., R.	2012	India	Theoretical	System development	Framework	None	Not specified/Does not apply	U-learning	WSNs, Mobile handheld devices	No	Picture, Lecture
Context's Personalized Recommendation for Improving Ubiquitous Learning in Health 2.0	Mendes Neto, F.M.A., Da Costa, A.L.L., Sombra, E.L.A.	2014	Brazil	Experimental	System development, Experimental execution	Model	Health	Not specified/Does not apply	U-learning	WSNs	No	none
Context Sensitive Content: Sensor Data for Computer-Supported Reflective Learning	Miller, L.A., Dettini, M.B., Mora, D.S., Rivera-Pelaez, V.A.	2015	Germany	Experimental	System development	Method	Health	Teritary	U-learning	WSNs, Mobile handheld devices	No	Learning-oriented environment
Design and implementation of a context-aware content service in mobile LMS	Ku, J.H.	2012	South Korea	Experimental	System development	Model	None	Not specified/Does not apply	U-learning	WSNs	No	Learning-oriented environment
Design of an Effective WSN-based Interactive U-Learning Model	Kim, H.-J., Cayless, R.D.B., Kim, T.-H.	2010	China	Theoretical	System development	Tool	None	Not specified/Does not apply	U-learning	WSNs	No	Learning-oriented environment
Development of a ubiquitous learning platform based on a real-time help-seeking mechanism	Hwang, G.-J., Wu, C.-H., Tseng, J.C.R., Huang, B.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	IT	Not specified/Does not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text
Smart Learning Systems Based on Social Media	Yoon, S., Song, S.M., Yoon, Y.-Y.	2011	South Korea	Experimental	System development, Experimental execution	Model	None	Not specified/Does not apply	U-learning	Mobile handheld devices	No	Learning-oriented environment

ESTUDOS MAIS RELEVANTES DO SM

Title	Author	Year	Country	Type of study	Methods used	Type of contribution	Science field where it is applied	Scholarly level of students	Type of learning environment	IoT technology types used	Does the authors recognize they use IoT explicitly in the paper?	Educational resources used
A Social Web of Things Approach to a Smart Campus Model	Allif, Y; Mathew, S.	2013	Sweden	Theoretical	Theory deep analysis	Framework	None	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Cloud-centered systems	Yes	Picture, Book/Text, Lecture, Video, Learning-oriented environment
A knowledge acquisition approach to developing Mindtools for organizing and sharing differentiating knowledge in a ubiquitous learning environment	Hwang, G.J; Chu, H.C; Lin, Y.S; Tsai, Chu, H.C; Hwang, G.J; Tsa, C.C.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Upper secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text
A knowledge engineering approach to developing mindtools for context-aware ubiquitous learning		2010	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Lower secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text
A Social Education Network Based on Location Sensing Information Using Smart Phones	Kang, J.M; Choi, S.Y.	2011	South Korea	Theoretical	System development	Tool	None	Not specifiedDoes not apply	U-learning	Mobile handheld devices	No	None
Advances in MASELTOV – Serious Games in a Mobile Ecology of Services for Social Inclusion and Empowerment of Recent Immigrants	Palaella, L; Dunwell, I; Gaved, M; Bobe	2013	Austria	Theoretical	System development	Tool	Linguistics	Not specifiedDoes not apply	U-learning	Mobile handheld devices	No	Book/Text, Learning-oriented environment, Serious games
An Adaptive and Personalized Ubiquitous Learning Middleware Support for Handicapped Learners	Boynbode, O; Bagula, A	2011	South Africa	Theoretical	System development	Middleware	None	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	No	None
An Adaptive System Supporting Collaborative Learning Based on a Location-Based Social Network and Semantic UserModeling	Choi, S.-Y. a., Kang, J.-M.b	2012	South Korea	Theoretical	System development	Tool	None	Not specifiedDoes not apply	U-learning	Mobile handheld devices	No	none
An EPC-based middleware enabling reusable and flexible mixed reality educational experiences	Fiori, A; Mariani, L; Patrono, L; Vergi	2013	Italy	Experimental	System development	Middleware	None	Post-secondary non-tertiary	M-learning	RFID tags, WSNs, Cloud-centered systems	Yes	Picture, Book/Text, Learning-oriented environment
An experience of collaboration using a PaaS for the smarter university model	Coccoli, M; Maresca, P; Stanganeli, L	2015	Italy	Experimental	System development, Experimental execution	Tool	None	Not specifiedDoes not apply	M-learning	Cloud-centered systems, Mobile handheld devices	Yes	Picture, Book/Text, Learning-oriented environment
Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system	Wang, S.L; Wu, C.Y.	2011	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	SCORM
Architecture for integrating real objects with Virtual Academic Communities	Vilanova, J.A.V; Solars, Z.M.A; Mar	2015	Colombia	Experimental	System development, Experimental execution	Architecture	IT	Bachelor or equivalent	U-learning	WSNs, Cloud-centered systems	Yes	Learning-oriented environment
Cloud E-learning for Mechantronics CLEM	Choi, K.M; James, A.E; Nanos, A.G; I	2015	United Kingdom	Experimental	System development, Experimental execution	Tool	IT	Bachelor or equivalent	E-learning	Cloud-centered systems	Yes	Learning-oriented environment
Computer Supported Ubiquitous Learning: Augmenting Learning Experiences in the Real World	Ogata, H	2008	Japan	Experimental	System development, Experimental execution	Tool	Language	Not specifiedDoes not apply	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Video, Learning-oriented environment, Dictionary
Conducting Situated Learning in a Context-Aware Ubiquitous Learning Environment	Wu, T.T; Yang, T.C; Hwang, G.J; Chu	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Lower secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Learning-oriented environment
Considering learning styles and context-awareness for mobile adaptive learning	Tortorella, R.A.W; Graf, S.	2015	Finland	Experimental	System development, Experimental execution	Tool	Physics	Post-secondary non-tertiary	U-learning	WSNs, Mobile handheld devices	No	Picture, Book/Text, Lecture, Video, Learning-oriented environment, Audio
Context-Aware Writing in Ubiquitous Learning Environments	Chen, T.S; Chang, C.S; Lin, J.S; Yu, I	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Language	Lower secondary	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Picture, Book/Text, Video, Learning-oriented environment
Criteria and Strategies of Ubiquitous Learning	Hwang, G.J	2006	Taiwan	Theoretical	Case study	Survey	Biology	Lower secondary	U-learning	RFID tags, WSNs, Mobile handheld devices	No	Learning-oriented environment
Design and Development of Digital Teaching Management System Based on Internet of Things	Xue, Y.-Fab., Sun, H.-L.c., Wu, Y.-H	2011	China	Experimental	System development, Experimental execution	Tool	None	Not specifiedDoes not apply	M-learning	RFID tags, Mobile handheld devices	Yes	Learning-oriented environment
Design and implementation of a context aware contents service model in mobile LMS	Ku, J.-H.	2014	South Korea	Theoretical	System development	Model	None	Not specifiedDoes not apply	U-learning	Mobile handheld devices	No	Learning-oriented environment
Development of Web-based Japanese Ministry and Oronotopozona Learning Assistant System with Sensor Network	Hsu, B; Ogata, H; Miyata, M; Li, M; Yq	2010	Japan	Experimental	System development, Experimental execution	Tool	Linguistics	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Cloud-centered systems, Mobile handheld devices	No	Book/Text, Dictionary
Energy efficient oriented mobile learning that emphasizes energy saving awareness education	Deikouk, A	2013	Algeria	Theoretical	System development	Tool	None	Not specifiedDoes not apply	M-learning	WSNs, Mobile handheld devices	Yes	Book/Text
Expanding the learning environment: combining physicality and virtuality	Domingo, M.G; Forner, J.A.M	2010	Spain	Experimental	Case study	Survey	None	Not specifiedDoes not apply	M-learning	None	Yes	None
Interacting with annotated objects in a Semantic Web of Things application	Coccoli, M; Torre, I.	2014	Italy	Theoretical	Theory deep analysis	Method	None	Not specifiedDoes not apply	E-learning	RFID tags, WSNs, Cloud-centered systems, Mobile handheld devices	Yes	none
Learning by Logging: Supporting Ubiquitous Learning Using a Logging Tool	Ogata, H; Li, M; Hsu, B; Usuki, N; Yq	2011	Japan	Experimental	System development, Experimental execution	Tool	None	Bachelor or equivalent	U-learning	Mobile handheld devices	Yes	Picture, Book/Text, Learning logs
Learning in a u-Museum: Developing a context-aware ubiquitous learning environment	Chen, C.G; Huang, T.C.	2012	Taiwan	Experimental	System development, Experimental execution	Tool	History	Upper secondary	U-learning	RFID tags, Mobile handheld devices	No	Picture, Book/Text, Audio
LORAMS: Sharing Learning Experiences with Social and Ubiquitous Media	Ogata, H; Misumi, T; Hsu, B; Li, M; Ei	2010	Japan	Experimental	System development, Experimental execution	Tool	None	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Video
M-learning Mode Research Based on Internet of Things	Yang, B; Nie, X; Shi, H; Gan, W	2011	China	Theoretical	System development	Tool	None	Not specifiedDoes not apply	M-learning	RFID tags, WSNs	Yes	None
Multimedia augmented m-learning: issues, trends and opportunities	Yusoffaz, A; Chang, V; Gan, A; Nee	2016	Malaysia	Experimental	Case study	Survey	None	Not specifiedDoes not apply	M-learning	Mobile handheld devices	Yes	none
Personalized English reading sequencing based on learning portfolio analysis	Wu, T.T; Huang, Y.M; Chao, H.C; Pai	2014	Taiwan	Experimental	System development, Experimental execution	Framework	Linguistics	Bachelor or equivalent	U-learning	RFID tags, Mobile handheld devices	No	Book/Text
Poster: Design and Implementation of IoT-enable Mobile E-Learning Platform for Medical Education	Chang, C.W; Lin, P.; Tseng, C.W; Kori	2015	Taiwan	Experimental	System development, Experimental execution	Tool	Health	Bachelor or equivalent	M-learning	WSNs	Yes	Picture, Learning-oriented environment, Question
RFID-based Ubiquitous Learning Environment for Outdoor Learning	Liu, T.Y; Chu, Y.L.; Tan, T.H; Chang, C	2007	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Upper secondary	U-learning	RFID tags, Mobile handheld devices	No	None
Smarter universities: A vision for the fast changing digital era	Coccoli, M; Guerio, A; Maresca, P; S	2014	Italy	Experimental	Case study	Survey	None	Not specifiedDoes not apply	E-learning	RFID tags, WSNs, Cloud-centered systems, Mobile handheld devices	Yes	none
Survey: Toward A Smart Campus Using the Internet of Things	Alghamdi, A; Shetty, S.	2016	USA	Experimental	Case study	Survey	None	Not specifiedDoes not apply	U-learning	Mobile handheld devices	Yes	Lecture, Learning-oriented environment
THE REMOTE EXPERIMENT COMPATIBILITY WITH INTERNET OF THINGS	Samola, C; Ursutiu, D; Jinga, V.	2016	Romania	Theoretical	Case study, Theory deep analysis	Theory review, Survey	None	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Cloud-centered systems, Mobile handheld devices	Yes	none
The role of big data and cognitive computing in the learning process	Coccoli, M; Maresca, P; Stanganeli, L	2016	Italy	Experimental	Case study	Survey	None	Not specifiedDoes not apply	U-learning	None	Yes	none
Towards a Multi-modal Emotion-awareness e-Learning System	Cabale, S.	2015	Spain	Theoretical	System development	Tool	None	Not specifiedDoes not apply	U-learning	WSNs	Yes	Learning-oriented environment, MOOC
Ubiquitous Learning: Teaching Modeling and Simulation with Technology	Moller, D.P.F; Haas, R; Vakizadzan, H	2013	Germany	Theoretical	Theory deep analysis	Systematic review/mapping	None	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Cloud-centered systems	Yes	Learning-oriented environment
Using the IOT to Construct Ubiquitous Learning Environment	Xue, R; Wang, L; Chen, J	2011	China	Theoretical	System development	Framework	None	Not specifiedDoes not apply	U-learning	RFID tags, WSNs, Mobile handheld devices	Yes	None
Using the RFIDs to Construct the Ubiquitous Self-Learning Environment for Understanding the Plants in the Schoolyard	Hsu, J.M; Lai, Y.S; Yu, P.T	2008	Taiwan	Experimental	System development, Experimental execution	Tool	Biology	Primary	U-learning	RFID tags, Mobile handheld devices	No	None
Context-Aware Support for Computer-Supported Ubiquitous Learning	Ogata, H; Yano, Y	2004	Japan	Experimental	System development	Tool	Language	Post-secondary non-tertiary	U-learning	RFID, PDA	No	Book/Text, Helper
A Model of Personalized Collaborative Computer Support Ubiquitous Learning Environment	El-Bilhouay, M.M; Ogata, H; Yano, Y	2008	Japan	Theoretical	System development	Tool	None	Not specifiedDoes not apply	U-learning	RFID tags	No	None