

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Uma abordagem de modelagem de conteúdo educacional
para plataforma de aprendizagem móvel baseada na
engenharia dirigida por modelos**

Marcus Vinícius Ribeiro de Carvalho

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

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marcus Vinícius Ribeiro de Carvalho

Uma abordagem de modelagem de conteúdo educacional
para plataforma de aprendizagem móvel baseada na
engenharia dirigida por modelos

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Ellen Francine Barbosa

USP – São Carlos
Setembro de 2021

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

C331a Carvalho, Marcus Vinícius Ribeiro de
Uma abordagem de modelagem de conteúdo
educacional para plataforma de aprendizagem móvel
baseada na engenharia dirigida por modelos / Marcus
Vinícius Ribeiro de Carvalho; orientadora Ellen
Francine Barbosa Barbosa. -- São Carlos, 2021.
200 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2021.

1. Aprendizagem móvel. 2. Modelagem de Conteúdo
Educativo. 3. Aplicações Educacionais Móveis,. 4.
Engenharia Dirigida por Modelos. 5. Engenharia de
Software. I. Barbosa, Ellen Francine Barbosa,
orient. II. Título.

Marcus Vinícius Ribeiro de Carvalho

**An educational content modeling approach for the
model-driven engineering-based mobile learning platform**

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Ellen Francine Barbosa

USP – São Carlos
September 2021

dedicatoria,

Dedico esse trabalho aos meus pais, Francisco Afonso e Maria Coeli. A minha esposa Luciana, sem seu apoio e compressão não teria consigo atingir esse objetivo. Aos meus filhos, Samuel e Germano.

AGRADECIMENTOS

Agradeço a Deus, por me manter firme nessa caminhada e sempre renovar minhas forças para superar os desafios apresentados para conclusão desse trabalho.

Agradeço a Ellen Francine Barbosa, minha orientadora, pela confiança, compreensão e paciência na orientação deste trabalho. A forma como conduziu a orientação desse trabalho foi um ensinamento que levo para minha vida acadêmica.

À minha esposa Luciana Coimbra, pela paciência e compressão durante toda essa caminhada. O seu suporte foi fundamental para conclusão desse etapa.

Aos meus filhos Samuel e Germano, por compreender os momentos ausentes durante a realização desse doutorado.

À meus pais, Francisco Afonso e Maria Coeli, pelo apoio que sempre me deram em todos os momentos de minha vida.

Aos colegas do Laboratório de Engenharia de Software do ICMC-USP (LabEs) e do Laboratório de Computação Aplicada à Educação e Tecnologia Social Avançada (CaEd) pela ajuda ao longo desse doutorado.

Aos professores do ICMC-USP, pelos ensinamentos e dedicação nas disciplinas do doutorado. Agradeço ainda, a todos funcionários ICMC-USP, pela atenção e dedicação que sempre me atenderam.

A todos os colegas do DINTER UESPI/USP, pelo companheirismo e incentivo durante a realização do doutorado.

À Universidade Estadual do Piauí (UESPI) pela oportunidade e apoio financeiro para conclusão desse doutorado.

*“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)*

RESUMO

CARVALHO, M. V. R. **Uma abordagem de modelagem de conteúdo educacional para plataforma de aprendizagem móvel baseada na engenharia dirigida por modelos.** 2021. 184 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

O cenário emergente do uso de ambientes computacionais para apoiar o processo de ensino e aprendizagem tem apresentado desafios aos profissionais desenvolvedores de aplicações e produtores de conteúdo educacional. Nesse contexto, a evolução da computação móvel contribuiu para o surgimento da modalidade de ensino e aprendizagem denominada de aprendizagem móvel (*m-learning*). A aprendizagem móvel faz uso da tecnologia móvel e de dispositivos móveis como meio de disseminação de conteúdo ao aprendiz. No entanto, a geração de conteúdo educacional para dispositivos móveis requer atenção especial a questões como: (i) restrições inerentes dos dispositivos móveis, como por exemplo a tela reduzida; (ii) e a complexidade de desenvolver aplicações móveis para a variedade de plataformas existentes. O presente trabalho de pesquisa propõe uma abordagem de modelagem para conteúdo educacional móvel, denominada EDUCOM.ML, que é apoiada pela MDE (*Model-Driven Engineering*) e a IMA-CID (*Integrated Modeling Approach - Conceptual, Instructional, Didactic*). EDUCOM.ML propõe uma abordagem de modelagem que envolve a criação de modelos responsáveis pela estruturação do conteúdo educacional, bem como por um conjunto de metamodelos que estruturam e modelam uma plataforma móvel. Além disso, dois protótipos foram desenvolvidos, o primeiro é a ferramenta web EDUCOM TOOL, que permite a elaboração de conteúdo educacional de um domínio do conhecimento, enquanto que o segundo é a aplicação educacional móvel EDUCOMAPP, que disponibiliza ao aprendiz o conteúdo educacional projetado na EDUCOM TOOL. Para verificar a percepção de utilidade da EDUCOM.ML por professores, foi elaborada uma avaliação e os resultados obtidos juntos aos participantes sugerem que a EDUCOM.ML pode contribuir no processo de elaboração de conteúdo educacional para aplicações educacionais móveis.

Palavras-chave: Ensino e Aprendizagem, Aprendizagem móvel, Aplicações Educacionais Móveis, Engenharia Dirigida por Modelos, Modelagem de Conteúdo Educacional.

ABSTRACT

CARVALHO, M. V. R. **An educational content modeling approach for the model-driven engineering-based mobile learning platform.** 2021. 184 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

The emerging scenario of the use of computational environments to support the teaching and learning process has presented challenges to application developers and educational content producers. In this context, the evolution of mobile computing contributed to the emergence of the teaching and learning modality called mobile learning (m-learning). Mobile learning makes use of mobile technology and mobile devices as a means of disseminating content to the learner. However, the generation of educational content for mobile devices requires special attention to issues such as: (i) inherent restrictions of mobile devices, such as the reduced screen; (ii) and the complexity of developing mobile applications for the variety of existing platforms. This research work proposes a modelling approach for mobile educational content, called EDUCOM.ML, which is supported by MDE (Model-Driven Engineering) and IMA-CID (Integrated Modeling Approach - Conceptual, Instructional, Didactic). EDUCOM.ML proposes a modelling approach that involves the creation of models responsible for structuring educational content, as well as a set of metamodels that structure and model a mobile platform. In addition, two prototypes were developed, the first is the web tool Educom Tool that allows the development of educational content from a knowledge domain, while the second is the mobile educational application EducomApp, which provides the learner with the educational content designed in Educom Tool To verify the perception of the usefulness of EDUCOM.ML by teachers, an evaluation was elaborated and the results obtained together with the participants suggest that EDUCOM.ML can contribute to the process of elaboration of educational content for mobile educational applications.

Keywords: Teaching and Learning, Mobile Learning, Mobile Educational Applications, Model-Driven Engineering, Educational Content Modeling..

LISTA DE ILUSTRAÇÕES

Figura 1 – Assinatura de banda larga móvel ativa por 100 habitantes.	29
Figura 2 – AIM-CID: estrutura geral.	37
Figura 3 – Passos para construção do Modelo Conceitual.	39
Figura 4 – Passo para construção do Modelo Instrucional.	41
Figura 5 – Passos para construção do Modelo Didático.	42
Figura 6 – Arquitetura de quatro camadas (BASSANI, 2012)	43
Figura 7 – <i>Framework</i> básico MDA (BASSANI, 2012)	44
Figura 8 – Processo de transformação na abordagem MDA (KLEPPE; WARMER; BAST, 2003)	45
Figura 9 – Abordagens de modelagem utilizadas na produção de conteúdo educacional	49
Figura 10 – Pontes entre espaços tecnológicos.	56
Figura 11 – Proposta de extensão da abordagem AIM-CID.	58
Figura 12 – Texto conciso, claro e objetivo facilita a visibilidade do conteúdo.	60
Figura 13 – Sugestão de tamanho de linha de texto.	61
Figura 14 – Uso de sensores dos dispositivos móveis.	62
Figura 15 – Imagem com estética e relevância.	63
Figura 16 – Ilustração e fotografia.	63
Figura 17 – Foco Icônico.	64
Figura 18 – Imagem que conta uma história. Disputa comercial entre nações.	64
Figura 19 – Resolução de imagem.	65
Figura 20 – Visão geral da EDUCOM.ML.	67
Figura 21 – Metamodelo Educom.	69
Figura 22 – Metamodelo Mobile.	76
Figura 23 – Metamodelo REACT-NATIVE.	79
Figura 24 – Interface para iniciar a modelagem de domínio do conhecimento da aplicação EDUCOM TOOL.	81
Figura 25 – Interface para definição de módulos, conceitos, itens de informação, elemen- tos instrucionais e modelo didático de um domínio.	82
Figura 26 – Componentes de implementação de instância da Educom.ml.	86
Figura 27 – Fomulário de definição de módulos do domínio.	87
Figura 28 – Fomulário de definição de conceitos.	88
Figura 29 – Atribuição de conceitos aos módulos de ensino.	89
Figura 30 – Fomulário de definição de Item de Informação - mídia texto.	90

Figura 31 – Fomulário de definição de Elemento Instrucional exemplo.	91
Figura 32 – Trecho do Modelo Educom instanciado para Lógica de Programação.	91
Figura 33 – Exemplo de Modelo Didático para o domínio de Lógica de Programação.	93
Figura 34 – Interfaces de definição do Modelo Didático.	94
Figura 35 – Regra de transformação dos Modelos da Educom para Modelo Mobile.	95
Figura 36 – Trecho de correspondências entre elementos do Modelo Educom e Modelo Mobile.	95
Figura 37 – Trecho dos modelos instanciados na abordagem de modelagem EDUCOM.ML.	97
Figura 38 – Transformação do Modelo React-native em código-fonte React-native.	98
Figura 39 – Telas iniciais do EDUCOMAPP.	99
Figura 40 – Algumas interfaces do EDUCOMAPP para do domínio de Lógica de programação.	100
Figura 41 – Exemplo de material didático (slide) analisado na atividade de avaliação da EDUCOM.ML.	106
Figura 42 – Distribuição de frequência de respostas sobre a formação acadêmica.	107
Figura 43 – Distribuição de frequência de respostas a categoria de entidade de trabalho.	108
Figura 44 – Distribuição de frequência de respostas a nível de ensino trabalhado.	108
Figura 45 – Distribuição de frequência de respostas a tempo de experiência no ensino.	109
Figura 46 – Frequência de respostas sobre experiência em produção de conteúdo educacional.	110
Figura 47 – Distribuição de frequência de respostas a uso de metodologia na produção de conteúdo educacional.	110
Figura 48 – Distribuição de frequência de respostas a facilidade de adaptação de material didático.	111
Figura 49 – Distribuição de frequência de respostas para familiaridade com TICs.	111
Figura 50 – Distribuição de frequência de respostas sobre interesse em uma abordagem de modelagem para <i>m-learning</i>	112
Figura 51 – Frequência de respostas sobre a utilidade da estruturação proposta pela EDUCOM.ML.	113
Figura 52 – Frequência de respostas sobre a utilidade da EDUCOM.ML no processo de produção de conteúdo.	113
Figura 53 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Conceitual da abordagem EDUCOM.ML.	114
Figura 54 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Instrucional da abordagem EDUCOM.ML.	114
Figura 55 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Didático da abordagem EDUCOM.ML.	115
Figura 56 – Frequência de respostas sobre a compreensão das diretrizes de produção de mídia da abordagem EDUCOM.ML.	115

Figura 57 – Frequência de respostas sobre o desejo de adoção da abordagem EDUCOM.ML no processo de produção de conteúdo educacional.	116
Figura 58 – Visão geral da distribuição de frequência relativas de respostas da avaliação de utilidade da proposta de abordagem modelagem EDUCOM.ML.	116
Figura 59 – Resumo de atividades desenvolvidas na pesquisa de doutorado.	122

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Método de correspondência de questões entre Modelo Educom e Modelo Mobile	96
Código-fonte 2 – Regras de transformação em Java - Modelo Educom	137
Código-fonte 3 – Regras de transformação em Java - Modelo Educom para Modelo Mobile	148
Código-fonte 4 – Regras de transformação em Java - Modelo Mobile para Modelo React-Native	158
Código-fonte 5 – Regras de transformação em Java - Modelo React-Native para código-fonte React-Native	165

LISTA DE TABELAS

Tabela 1 – Critérios de inclusão e exclusão de publicações.	47
Tabela 2 – Número de trabalhos recuperados por base indexada.	48
Tabela 3 – Trabalhos selecionados	49
Tabela 4 – Resumo da respostas questões de pesquisa.	51
Tabela 5 – Diretrizes para produção de mídia.	66
Tabela 6 – Classe IKnowledgeDomain	68
Tabela 7 – Classe Module	70
Tabela 8 – Classe Image	73
Tabela 9 – Classe Video	73
Tabela 10 – Classes que suportam a definição de parâmetro de avaliação.	74
Tabela 11 – Classe Image	77
Tabela 12 – Classe Video	77
Tabela 13 – Classe Audio	77
Tabela 14 – Perfil de idade dos participantes da avaliação.	107

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Contextualização e motivação	27
1.2	Objetivos	31
1.3	Organização da Tese	32
2	FUNDAMENTAÇÃO TEÓRICA	33
2.1	Aprendizagem móvel (<i>m-learning</i>)	33
2.2	Modelagem de conteúdo educacional com AIM-CID	37
2.3	Engenharia Dirigida por Modelos (<i>Model-driven engineering – MDE</i>)	42
2.3.1	<i>Model driven architecture (MDA)</i>	43
2.4	Modelagem de conteúdo educacional: um mapeamento sistemático	46
2.4.1	<i>Metodologia, planejamento e execução</i>	46
2.4.2	<i>Processo de seleção dos estudos primários</i>	47
2.4.3	<i>Seleção final e extração do resultados</i>	48
2.4.4	<i>Análise dos trabalhos selecionados</i>	48
2.5	Considerações finais	52
3	ABORDAGEM DE MODELAGEM DE CONTEÚDO EDUCACIONAL PARA M-LEARNING	55
3.1	Visão Geral	55
3.2	Abordagem Educom.ml	57
3.2.1	<i>Modelo Conceitual da Educom.ml</i>	57
3.2.2	<i>Modelo Instrucional da Educom.ml</i>	59
3.2.3	<i>Design de mídia para dispositivos móveis - Media Model</i>	59
3.2.3.1	<i>Texto</i>	59
3.2.3.2	<i>Imagem</i>	62
3.2.3.3	<i>Vídeo</i>	64
3.3	Modelagem do conteúdo educacional	67
3.3.1	<i>Metamodelo Educom</i>	68
3.4	Modelagem da Plataforma Alvo	75
3.4.1	<i>Metamodelo Mobile</i>	75
3.4.2	<i>Metamodelo React-native (MRn)</i>	77
3.5	Ferramenta de autoria Educom Tool	80

3.6	Considerações finais	83
4	APLICANDO A ABORDAGEM DE MODELAGEM DE CONTEÚDO EDUCACIONAL PARA M-LEARNING	85
4.1	Modelagem do conteúdo educacional	85
4.1.1	<i>Instanciando o Modelo Conceitual (MC)</i>	87
4.1.2	<i>Instanciando Modelo Instrucional (MI)</i>	89
4.1.3	<i>Instanciando Modelo Didático</i>	91
4.1.4	<i>Instanciando Modelo Mobile</i>	93
4.1.5	<i>Instanciando Modelo React-native (MRC)</i>	96
4.1.6	<i>Aplicativo móvel gerado</i>	98
4.1.7	<i>Considerações finais</i>	100
5	AVALIAÇÃO DA EDUCOM.ML: PROVA DE CONCEITO COM O PROFESSOR	103
5.1	Planejamento, metodologia e execução	103
5.1.1	<i>Formulação das hipóteses</i>	104
5.1.2	<i>Instrumentação</i>	104
5.1.3	<i>Execução da avaliação</i>	105
5.2	Análise e interpretação de resultados	106
5.3	Riscos à Validade da Avaliação	117
5.4	Considerações Finais	119
6	CONCLUSÕES	121
6.1	Contribuições da pesquisa	123
6.2	Limitações da pesquisa	124
6.3	Trabalhos futuros	125
6.4	Publicações	126
6.4.1	<i>Publicada</i>	126
6.4.2	<i>Submetida</i>	126
	REFERÊNCIAS	127
APÊNDICE A	IMPLEMENTAÇÃO DE REGRAS DE TRANSFORMAÇÃO DE MODELOS	137
A.1	Instância do Modelo Educom	137
A.2	Instância do Modelo Mobile	148
A.3	Instância do Modelo React Native	158
A.4	Gerar código-fonte React Native	165
APÊNDICE B	INSTRUMENTAÇÃO DA AVALIAÇÃO	177

B.1	Questionário de avaliação da Educom.ml	177
B.2	Questionário de perfil do participante	180

INTRODUÇÃO

Este capítulo apresenta o contexto no qual o doutorado está inserido, além de evidenciar sua motivação e os objetivos que se pretende alcançar. É apresentada ainda, a organização do trabalho proposto.

1.1 Contextualização e motivação

O avanço das Tecnologias da Informação e Comunicação (TIC's) tem proporcionado varias iniciativas para melhorar e estimular o processo de aprendizagem (BARBOSA; MALDONADO, 2006)(MARCOLINO; BARBOSA, 2016)(BIARD; COJEAN; JAMET, 2017)(FU; HWANG, 2018)(JAHNKE; LIEBSCHER, 2020). No entanto, incluir recursos tecnológicos no processo de educação não garante a melhora do aprendizado (JAHNKE; LIEBSCHER, 2020). O cenário da educação mediada por TICs apresenta aos profissionais da computação a demanda de preparar uma estrutura computacional que forneça aos cursos mecanismos para estruturar, administrar, apresentar conteúdo educacional e acompanhar a evolução do aprendiz (KENSKI, 2007) (WASTIAU *et al.*, 2013) (BRAGA *et al.*, 2014) (MARCOLINO; BARBOSA, 2016) (BIARD; COJEAN; JAMET, 2017) (RODRÍGUEZ; RIAZA; GÓMEZ, 2017) (MAYER; FIORELLA; STULL, 2020) (CURUM; KHEDO, 2020) (DANISH; HMELO-SILVER, 2020). Segundo Barbosa, Borges e Maldonado (2013), esta estrutura consiste em material didático (livros, artigos, notas de aulas, áudio, vídeo) e infraestrutura (recursos computacionais e tecnológicos), denominada de módulo. Barbosa, Borges e Maldonado (2013).

A modelagem de material didático de qualidade é uma tarefa essencial no desenvolvimento de módulos educacionais, sendo considerada uma atividade complexa que exige a observação de fatores como: especificação de aspectos característicos do domínio do conhecimento; a definição de atividades práticas e sua avaliação; e o estabelecimento de ordem pedagógica lógica de apresentação do material didático modelado (BORGES, 2010) (JAHNKE; LIEBSCHER, 2020). Apesar da importância da estruturação do material didático, pesquisas

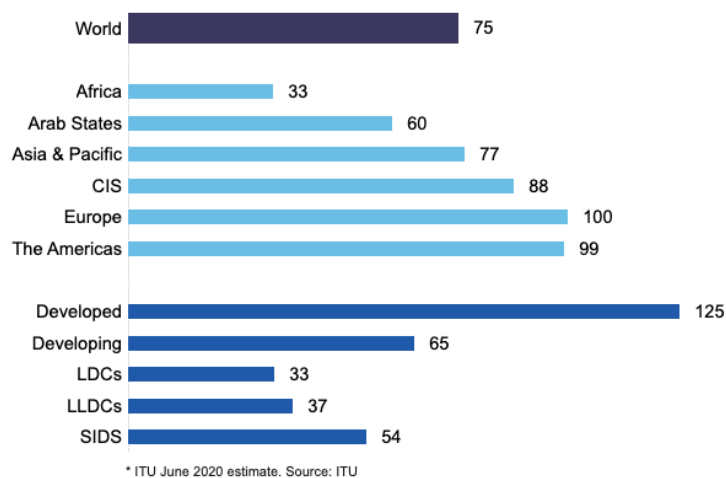
estão concentradas mais nas questões de gerenciamento do ensino (automatização do processo de disponibilização de material didático, acompanhamento da aprendizagem, adaptação de conteúdo a diferente modalidade de ensino e aprendizagem) (KEMCZINSKI *et al.*, 2012) (CHIMALA-KONDA; NORI, 2013) (Júnior *et al.*, 2014) (BANSAL; DALRYMPLE, 2016) (LOUHAB; BAHNASSE; TALEA, 2018) (ENNOUAMANI; MAHANI; AKHARRAZ, 2020). Portanto, a condução de pesquisas relacionadas com a modelagem de conteúdo educacional representa uma atividade essencial para a estruturação do conhecimento, contribuindo assim com o sucesso e efetividade da aprendizagem.

À medida que a tecnologia é incorporada ao processo de aprendizagem, novos desafios e adequações do conteúdo educacional são necessários para atender suas características (PRIETO; MIGUELÁÑEZ; GARCÍA-PEÑALVO, 2013)(AL-AMRI *et al.*, 2020). Um exemplo dessa evolução tecnológica é o uso da tecnologia móvel no processo de ensino e aprendizagem. A tecnologia móvel apresenta ao aprendiz uma realidade diferente de comunicar-se e adquirir o conhecimento. Sánchez, Cortijo e Javed (2014) classifica esse aprendiz como "nativo digital", isto é, são pessoas que nasceram imersos na tecnologia digital. Sánchez, Cortijo e Javed (2014), ainda, sugerem que professores precisam utilizar estratégias de ensino que atendam as necessidades de aprendizagem desse novo perfil de aprendiz. No contexto da educação, a tecnologia móvel proporcionou o surgimento da aprendizagem móvel (*m-learning*), que se caracteriza como uma modalidade de ensino e aprendizagem que fornece ao aprendiz a possibilidade de estudo sem um lugar fixo ou momento pré-determinado (SRIVASTAVA; GULATI, 2014) (BORGES, 2010) (ENNOUAMANI; MAHANI; AKHARRAZ, 2020).

A atenção com *m-learning* vem crescendo com a popularização da computação móvel. As tecnologias móveis podem ser consideradas atualmente como a tecnologia da informação e comunicação mais popular entre as pessoas. Segundo o relatório 2020 *Mobile Industry Impact* (INTELLIGENCE, 2020), elaborado pelo *Groupe Speciale Mobile Association - GSMA*, em 2019 66% da população mundial utilizava a tecnologia móvel para se comunicar. No contexto da educação, 40% da população mundial utilizou a tecnologia móvel com o objetivo de se manter informado. Olhando para o contexto nacional, a Pesquisa Nacional por Amostra de Domicílios (PNAD) 2018 (IBGE, 2018), levantou que 98% das pessoas pesquisadas recorriam ao celular para acessar internet, enquanto 50,7% usavam o computador. Outra pesquisa realizada pela *International Telecommunication Union-ITU* apresenta dados sobre a quantidade de assinaturas de banda larga móvel ativa por 100 habitantes. A Figura 1 sintetiza os resultados da pesquisa demonstrando que mundialmente a cada 100 habitantes 75 possuem assinatura de banda larga ativa. Esses dados demonstram o potencial que dispositivos móveis oferecem para ajudar o processo de ensino e aprendizagem, principalmente em áreas que não existe a disponibilidade de internet banda larga fixa.

De fato, a *m-learning* flexibiliza o momento e lugar da aprendizagem, porém, para manter o aprendiz engajado na atividade de aprendizagem o conteúdo educacional precisa ser

Figura 1 – Assinatura de banda larga móvel ativa por 100 habitantes.



Fonte: International Telecommunication Union (ITU) 2020 – www.itu.int.

preparado considerando os recursos e as limitações dos dispositivos móveis (celular, *smartphone*, *tablet*) (HEFLIN; SHEWMAKER; NGUYEN, 2017) (CURUM; KHEDO, 2020) (JAHNKE; LIEBSCHER, 2020) (ENNOUAMANI; MAHANI; AKHARRAZ, 2020). Vale lembrar que o conteúdo educacional preparado para o ensino formal não considera aspectos como a variedade de modelos de dispositivos móveis existentes, bem como os diferentes recursos e tamanhos de telas disponíveis (GIPPLE; LORD, 2016).

Segundo Duarte e Barbosa (2013a) a produção de conteúdo educacional para *m-learning* envolve a modelagem de requisitos como: i) funcionalidade, dedicando atenção ao uso de recursos multimídia; ii) desempenho, observando questões como consumo de banda e tempo de carregamento do conteúdo; e iii) usabilidade, construindo interfaces fáceis de aprender, reconhecer e lembrar. Outro aspecto considerado no uso de *m-learning* no processo de aprendizagem está relacionado à formação de professores para uso de TIC's. Pesquisas demonstram que professores sentem dificuldades para incorporar TIC's em suas aulas (FARJON; SMITS; VOOGT, 2019) (TONDEUR; HOWARD; YANG, 2021) (RUBACH; LAZARIDES, 2021) (WANG; LU, 2021) (HOWARD *et al.*, 2021). Logo, uma proposta de modelagem com ferramentas e orientações de produção de conteúdo educacional para dispositivos móveis pode incentivar professores a adotar *m-learning* no processo de ensino e aprendizagem.

No desenvolvimento de aplicações móveis, além da atenção com a produção do conteúdo educacional, existem preocupações de implementação que precisam ser consideradas. O desenvolvimento de aplicação educacional móvel enfrenta os mesmos desafios apresentados na construção de aplicações móveis para outras áreas. Portanto, desafios como heterogeneidade de sistemas operacionais, linguagem de desenvolvimento distinta para cada plataforma, definição de interface de usuário e uso de recursos físicos são preocupações também presentes no desenvolvimento de aplicações educacionais móveis (AHMAD *et al.*, 2018) (MARTINEZ *et*

al., 2020). Destarte, em linhas gerais, entende-se que projetos de implementação de aplicações educacionais móveis leva consideração dois aspectos: i) como o conteúdo educacional deve ser estruturado para atender as características de dispositivos móveis? ii) como enfrentar questões como diversidade de dispositivos e sistemas operacionais, as diversas linguagens de programação (nativa, híbrida, plataforma-cruzada) disponíveis para implementação e como ofertar uma visão unificada (produtores de conteúdo e desenvolvedores) de projeto no desenvolvimento de aplicações educacionais móveis? Visando responder essas questões de estruturação do conteúdo educacional e do desenvolvimento de aplicações móveis, este trabalho de pesquisa de doutorado adotou duas abordagens:

- A primeira questão foi enfrentada com a adaptação da abordagem AIM-CID (Abordagem Integrada de Modelagem — Conceitual, Instrucional e Didática) para *m-learning*. A abordagem AIM-CID apresenta uma modelagem de conteúdo educacional que fornece aos profissionais da educação um guia de produção de material didático, contemplando a definição de conceitos do domínio de conhecimento, a elaboração de elementos instrucionais e o sequenciamento da apresentação do conteúdo didático. A abordagem AIM-CID é genérica, isto é, o material didático modelado não é direcionado a uma modalidade de ensino e aprendizagem específica, portanto foi necessário incluir sugestões de adaptação da abordagem a modalidade de aprendizagem móvel. Pesquisas aplicando a AIM-CID sugerem que a abordagem tem contribuído para melhorar estruturação do conhecimento, minimizando a complexidade de modelagem de conteúdo educacional (BARBOSA; MALDONADO, 2004) (BARBOSA; MALDONADO, 2006) (BORGES, 2010) (SILVA; BARBOSA; MALDONADO, 2011) (MACHADO *et al.*, 2012) (FILHO; CRISTIANO, 2016).
- Os desafios da implementação de aplicações móveis foi enfrentado com a abordagem da Engenharia Dirigida por Modelos (*Model Driven Engineering* – MDE) para desenvolvimento de software. A MDE pode minimiza os desafios de heterogeneidade de tecnologias no desenvolvimento de aplicações, além de possibilitar aos interessados no projeto uma visão unificada dos requisitos da aplicação (IVANOV; BÉZIVIN; AKSIT, 2002)(BÉZIVIN, 2006) (ROCCO *et al.*, 2015) (Latif *et al.*, 2017) (BUCCHIARONE *et al.*, 2020). A MDE permite o desenvolvimento de sistemas complexos definindo modelos que estão distantes da tecnologia de implementação e mais próximos do domínio do problema (BÉZIVIN, 2006) (WHITTLE; HUTCHINSON; ROUNCFIELD, 2014) (CARVALHO; LOPES; ABDELOUAHAB, 2015) (OLIVEIRA *et al.*, 2015) (ROCCO *et al.*, 2015). Além disso, a MDE tem o foco do desenvolvimento na geração de artefatos por meio da transformação de modelos (WHITTLE; HUTCHINSON; ROUNCFIELD, 2014) (HRYCYK; SOARES; AGNER, 2018). Em geral, um modelo representa uma visão parcial e simplificada de um sistema, sendo necessário a criação de diferentes modelos para entender o sistema por completo (ALBERTO, 2015).

1.2 Objetivos

Considerando a motivação e contextualização apresentada, este projeto de doutorado tem como principal objetivo propor uma abordagem de modelagem de conteúdo educacional para *m-learning* (EDUCOM.ML — *Educational Content Modeling for M-Learning*). A EDUCOM.ML apoia a produção do conteúdo didático e o desenvolvimento de aplicações móveis estabelecendo um conjunto de modelos que auxiliam professores na produção do conteúdo didático e desenvolvedores na criação de aplicações educacionais móveis. Considerando que um dos produtos gerados é uma aplicação educacional móvel, a proposta também contribui com o processo de aprendizagem de aprendizes.

Nessa perspectiva, a EDUCOM.ML busca obter como resultado os seguintes artefatos:

1. Um conjunto de metamodelos que estrutura o conteúdo educacional de um domínio do conhecimento, cujo objetivo é sua apresentação em uma aplicação móvel;
2. Diretrizes que auxiliam os produtores de conteúdo (professores, tutores, autores) na criação de mídias como texto, imagem e vídeo preparados para aplicações móveis;
3. Um conjunto de metamodelos destinados a modelar a plataforma de desenvolvimento de aplicações móveis;
4. Uma ferramenta web de autoria de conteúdo educacional implementada conforme a abordagem EDUCOM.ML;
5. Por fim, a aplicação móvel gerada com o conteúdo educacional modelado com a abordagem EDUCOM.ML.

Vale destacar que o processo de geração do conteúdo educacional é apoiado por uma proposta de extensão da abordagem AIM-CID e pelo uso da abordagem MDE. Desde modo, percebe-se que o trabalho desenvolvido no doutorado preocupou-se tanto com os produtores de conteúdo educacional, como também em auxiliar a equipe de desenvolvimento de aplicações móveis.

A fim de alcançar o objetivo principal da proposta de doutorado, um conjunto de atividades foram executadas considerando as seguintes fases: i) Fase de embasamento teórico; ii) Fase de projeto e modelagem; iii) Fase de implementação; iv) Fase de avaliação da proposta.

- Com respeito ao embasamento teórico necessário para o desenvolvimento do trabalho, foram estudados os seguintes temas: i) aprendizagem móvel; ii) abordagem de modelagem AIM-CID; iii) abordagem de modelagem MDE (*Model-Driven Engineering*); iv) melhores práticas de design de mídias.

- Com respeito à execução do projeto foram propostos: i) metamodelo para a AIM-CID; ii) metamodelos para apresentação de conteúdo educacional em aplicações móveis; iii) metamodelo para a linguagem de programação móvel utilizada na implementação da aplicação educacional móvel;
- Quanto a aspectos de desenvolvimento, foram realizadas as seguintes implementações: i) um conjunto de regras de transformações que suportam a geração do conteúdo educacional modelado para aprendizagem móvel; ii) uma ferramenta web de autoria de conteúdo educacional, denominada *Educom Tool*; iv) uma aplicação educacional móvel para apresentação de conteúdo educacional, denominada *EducomApp*.
- Quanto à fase de avaliação, foi realizada uma avaliação da abordagem de modelagem de conteúdo para *m-learning*. A avaliação apontou evidências da contribuição da EDUCOM.ML no processo de produção de conteúdo educacional e implementação de aplicativo educacional móvel.

1.3 Organização da Tese

Neste capítulo foram apresentados a motivação e o contexto em que esta pesquisa se insere, bem como os objetivos pretendidos do projeto de doutorado. O restante do documento é organizado como se segue.

No Capítulo 2 é apresentada a teórica estudada para fundamentar a proposta de abordagem de modelagem de conteúdo educacional para aprendizagem móvel. Apresentando os conceitos principais da abordagem AIM-CID e da Engenharia Dirigida por Modelos (MDE).

No Capítulo 3 é apresentada a abordagem de modelagem de conteúdo educacional para *m-learning*, com o detalhamento dos metamodelos responsáveis pela estruturação do conteúdo didático e pela implementação da aplicação educacional móvel.

O Capítulo 4 apresenta a abordagem de modelagem aplicada a um domínio do conhecimento, com a criação de instância dos metamodelos da EDUCOM.ML que estruturam o conteúdo e implementam uma aplicação educacional móvel.

No Capítulo 5 são discutidas a condução e avaliação da proposta de modelagem, identificando evidências de seu benefício na produção de conteúdo educacional para aplicações móveis.

No Capítulo 6 são apresentadas as conclusões do trabalho de doutorado, com suas contribuições e as propostas de trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Modalidades de aprendizagem atuais, como o ensino à distância, colocam desafios motivadores para pesquisadores da área de educação e de Tecnologias da Informação e Comunicação (TICs). Os pesquisadores da educação precisam propor novas abordagens pedagógicas que aproveitem os recursos oferecidos pelas TICs no processo de ensino e aprendizagem de forma adequada, enquanto a equipe de TI deve concentrar esforços para atender os novos requisitos pedagógicos e tecnológicos. (BORGES, 2010) (BARBOSA; BORGES; MALDONADO, 2013) (BRAGA; PIMENTEL; DOTTA, 2013) (SARRAB *et al.*, 2018).

De fato, a variedade de recursos e funcionalidades fornecidos pelas TICs modificam a forma de interação entre aprendizes e professores, possibilitando que a aprendizagem ultrapasse os limites da sala de aula, de modo colaborativo e flexível (MARCOLINO; BARBOSA, 2016) (JIUGEN; RUONAN, 2016) (SANTOS *et al.*, 2016). Além disso, o conteúdo didático preparado para uma determinada TIC pode não ser adequado para outro (SARRAB *et al.*, 2018).

Neste capítulo discorre-se sobre a fundamentação teórica dos principais conceitos que são abordados neste trabalho de doutorado. A Seção 2.1 são apresentados os conceitos relacionados à aprendizagem móvel (*m-learning*). A Seção 2.2 apresenta a AIM-CID, que é uma abordagem genérica para modelagem de conteúdo educacional. Finalmente, na Seção 2.3 são descritos os conceitos e benefícios do uso da abordagem da Engenharia Dirigida por Modelo (MDE) no processo de desenvolvimento de software.

2.1 Aprendizagem móvel (*m-learning*)

A tecnologia móvel vem modificando a forma como tarefas cotidianas estão sendo realizadas. Atualmente, realizar uma compra ou utilizar os serviços bancários são atividades que podem ser realizadas por meio da tecnologia móvel a qualquer tempo e lugar. Essa característica

de mobilidade fornecida pela tecnologia móvel é um recurso que pode beneficiar a tarefa de aprendizagem. Segundo [Castrillo, Martín-Monje e Bárcena \(2014\)](#), com uma sociedade “em movimento” e constantemente conectada graças à tecnologia de seus dispositivos móveis e da velocidade de suas redes sem fio, é natural que *tablets*, PCs (*Personal Computers*) e *smartphones* também sejam utilizados na educação. [Heflin, Shewmaker e Nguyen \(2017\)](#) sugerem que a tecnologia móvel fornece aos educadores a oportunidade de repensar o ensino e a aprendizagem.

Aprendizagem móvel ou *m-learning* é a modalidade de ensino e aprendizagem mediada por ferramentas da tecnologia móvel. [Martin e Ertzberger \(2013\)](#) conceituam *m-learning* como a aprendizagem que ocorre quando o aprendiz tem acesso à informação a qualquer hora e em qualquer lugar utilizando tecnologias móveis para realizar atividades autênticas no contexto de sua aprendizagem. De acordo com [Chavoshi e Hamidi \(2019\)](#), *m-learning* pode ser definido como obter conhecimento e habilidade por meio dos benefícios do uso das tecnologias móveis, em qualquer momento e lugar, levando à mudança de comportamento de aprendizagem. Para [Danish e Hmelo-Silver \(2020\)](#) a aprendizagem móvel é aprender considerando vários contextos, com auxílio de interações sociais, com uso de dispositivos eletrônicos pessoais que permitem capturar imediatamente informações ou fornecer informações ao aprendiz. A Organização das Nações Unidas para a Educação, a Ciência e a Cultura (UNESCO) também conceituou aprendizagem móvel, definindo-a como o uso de tecnologias móveis, isoladamente ou em combinação com outras TICs, de modo a permitir a aprendizagem a qualquer hora e em qualquer lugar ([UNESCO, 2014](#)).

Analisando os conceitos apresentados, observa-se que a aprendizagem móvel é ancorada no uso de dispositivos pessoais (*smartphone*, *tablet*), na ausência de tempo e local determinado para o aprendiz estudar e na capacidade de interatividade dos dispositivos móveis (interatividade com o dispositivo móvel e com outros agentes no processo de ensino e aprendizagem).

O potencial do uso da tecnologia móvel como uma ferramenta no processo de ensino e aprendizagem chamou atenção de organizações como a UNESCO. A UNESCO publicou um documento intitulado “Diretrizes de políticas para a aprendizagem móvel” ([UNESCO, 2014](#)), o qual destina-se aos formuladores de políticas públicas e recomenda dez ações que agentes públicos devem observar na adoção da aprendizagem móvel. Resumidamente algumas dessas ações são destacadas:

1. **Treinar professores sobre como fazer avançar a aprendizagem mediante tecnologias móveis:** pesquisas da UNESCO mostraram que, sem orientação e capacitação, os professores frequentemente utilizam a tecnologia para “fazer coisas antigas de formas novas”, ao invés de transformar e melhorar abordagens de ensino e aprendizagem.
2. **Fornecer apoio e formação a professores por meio de tecnologias móveis:** a UNESCO verificou que poucos sistemas educacionais utilizam tecnologias móveis como apoio ao trabalho e ao desenvolvimento de professores, de forma especial para os que trabalham

em áreas distantes e com recursos escassos.

3. **Criar e aperfeiçoar conteúdos educacionais para uso em aparelhos móveis:** atualmente, a maioria dos conteúdos educacionais, incluindo os materiais digitais, não estão disponíveis para aparelhos móveis, nem aproveitam integralmente as propriedades específicas de multimídia, de comunicação e, às vezes, de localização desses dispositivos.
4. **Ampliar e melhorar as opções de conectividade, assegurando também a equidade:** um estudante que não pode usar uma rede móvel, seja por razões econômicas ou geográficas, tem seu acesso negado a uma impressionante e crescente gama de oportunidades de aprendizagem.

Refletindo sobre as diretrizes recomendadas pela UNESCO, é possível identificar algumas questões que precisam de investigação por parte da academia. As questões relacionadas com a preparação de professores e com o aperfeiçoamento de conteúdo educacional apresentado ao aprendiz são temas motivadores para os pesquisadores. Em seu estudo [Heflin, Shewmaker e Nguyen \(2017\)](#) sugerem que o uso de dispositivos móveis e software educacional, em geral, é mais eficaz quando os professores conseguem projetar a experiência de aprendizagem combinando dispositivo/programa com os objetivos da experiência educacional.

No estudo de [Iqbal e Bhatti \(2020\)](#), professores relatam acreditar no ensino e aprendizagem por meio de dispositivos móveis, porém expressam preocupação com possíveis distrações no uso de *smartphone*, bem como com ausência de capacitação, conhecimento técnico para adoção de *m-learning* no cenário de ensino superior. Portanto, nesse contexto, pode-se inferir que para um efetivo uso de *m-learning* os professores necessitam ter domínio da tecnologia móvel, e também, o conteúdo didático preparado por eles precisa aproveitar os recursos oferecidos pelos dispositivos móveis (recursos multimídia e de interatividade). Essas dificuldades de domínio da tecnologia e da produção de conteúdo didático podem ser minimizadas com a oferta de abordagens de modelagem destinadas à estruturação do conteúdo educacional, além de ferramentas que auxiliem a produção de material didático considerando as características da aprendizagem móvel ([BORGES, 2010](#)) ([BARBOSA; BORGES; MALDONADO, 2013](#)).

No contexto do desenvolvimento de sistemas de aprendizagem móvel, a literatura tem apresentado estudos que especificam um conjunto de características desejadas ([HWANG; WU; KE, 2011](#)) ([OZDAMLI; CAVUS, 2011](#)) ([HASHEMI et al., 2011](#)) ([ALALWAN; ALZHRANI; SARRAB, 2013](#)). Uma síntese dessas características são apresentadas a seguir:

- Fornecer ao estudante a escolha de onde e quando estudar, isto é, a atividade de estudo deixa de se limitar ao espaço formal da sala de aula;
- Aproveitar os recursos de comunicação síncrona e assíncrona dos dispositivos móveis para melhorar a interação do estudante com os professores e outros estudantes;

- Aproveitar melhor o tempo ocioso do estudante, permitindo aprendizagem durante atividades de deslocamento ou do espaço escolar;
- Facilitar a coleta de informações no processo de ensino e aprendizagem por meio de captura de texto, áudio, vídeo e sensores.
- Fornecer menor custo, já que dispositivos móveis possuem preços mais acessíveis que computadores. Existe ainda o menor custo de serviços de acesso à internet;

Outro estudo realizado por [Duarte e Barbosa \(2013\)](#), investiga os requisitos que devem ser considerados no desenvolvimento de ambientes de aprendizagem móvel. [Duarte e Barbosa \(2013\)](#) apresentaram um catálogo de requisitos para modalidade *m-learning* que aborda aspectos técnicos e, também, aspectos relacionados às questões educacionais, econômicas e sócio-culturais. O catálogo é organizado em quatro eixos:

- Critérios técnicos: preocupam-se com as questões de funcionalidade, segurança e desempenho. Os critérios de funcionalidade tratam da oferta de atividades relacionadas com o ambiente educacional oferecido a estudantes e professores. Os critérios de segurança estão relacionados com a confiabilidade, disponibilidade e integridade ao acessar o ambiente educacional. Os critérios de desempenho estão relacionados com o consumo de bateria, capacidade de processamento, tempo de resposta, expansibilidade e escalabilidade, quantidade de memória livre e capacidade de entrada/saída.
- Critérios educacionais: preocupam-se em promover um ambiente para facilitar a aprendizagem flexibilizando atividades e tarefas de estudantes e professores. Os critérios consideram aspectos pedagógicos e de usabilidade.
- Critérios sócio-culturais: levam em consideração a comunicação e portabilidade do ambiente, observando fatores relacionados com o nível cultural dos indivíduos.

Apesar das perspectivas promissoras do uso da aprendizagem móvel no processo de ensino e aprendizagem, ainda existem questões que precisam de soluções. [Alalwan, Alzahrani e Sarrab \(2013\)](#) destacam algumas dessas questões: i) restrição de tamanho de tela; ii) busca pela melhor infraestrutura para ser utilizada em ambientes de aprendizagem móvel; iii) propostas de interfaces universais, além da restrição de armazenamento de dados; iv) não garantia que haverá sempre conexão disponível; v) *design* de aplicações móveis sensíveis ao contexto; vi) necessidade de adequação de ambiente educacional a diferentes plataformas (IOS, Android, Windows); vii) professores, instrutores e aprendizes devem ser capacitados para o uso do ambiente móvel.

A modalidade de ensino e aprendizagem móvel é promissora, porém para que professores e aprendizes usufruam de seus benefícios, questões técnicas e pedagógicas precisam de enfrentamento. Considerando o conteúdo didático, pesquisas estão sendo conduzidas para sistematizar

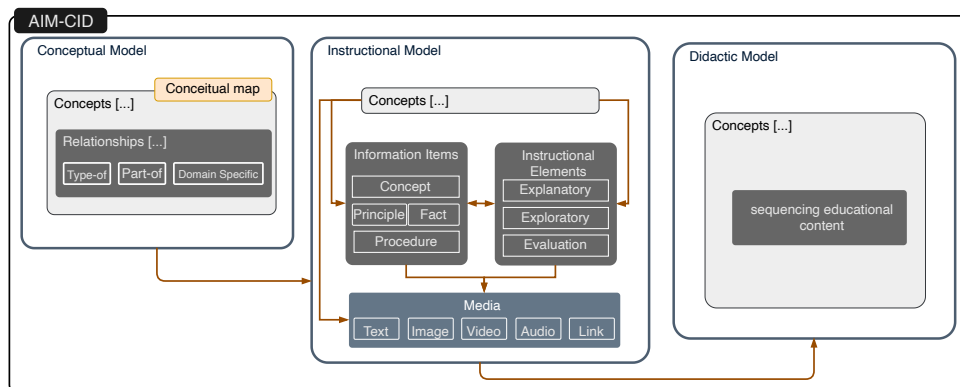
a estruturação de conteúdo educacional, porém não direcionadas a modalidade *m-learning* (BORGES, 2010) (BARBOSA; BORGES; MALDONADO, 2013) (BRAGA; PIMENTEL; DOTTA, 2013) (DUARTE; BARBOSA, 2013) (FILHO; CRISTIANO, 2016) (ENNOUAMANI; MAHANI; AKHARRAZ, 2020). Como em toda modalidade de ensino e aprendizagem, o conteúdo educacional de qualidade é uma parte importante para a efetiva compreensão de um domínio do conhecimento. Portanto, na aprendizagem móvel é preciso que professores estejam conscientes que o material didático precisa ser produzido considerando as restrições e os recursos fornecidos por dispositivos móveis (PAULINS; BALINA; ARHIPOVA, 2015)(CHAVOSHI; HAMIDI, 2019).

2.2 Modelagem de conteúdo educacional com AIM-CID

Barbosa e Maldonado (2004) propuseram a Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática (AIM-CID), que tem como finalidade a estruturação genérica de conteúdo educacional. Em linhas gerais, o conteúdo educacional é uma composição de conceitos, relacionamentos entre conceitos e informações adicionais como exercícios, avaliações, exemplos, entre outros elementos complementares (BARBOSA; MALDONADO, 2004).

Considerando essa visão de composição do conteúdo educacional, a AIM-CID propõe uma modelagem baseada em três perspectivas, como ilustra a Figura 2. A primeira perspectiva trata de modelar os conceitos relevantes do domínio de conhecimento e seus inter-relacionamentos, denominada Modelo Conceitual. A segunda perspectiva trata de modelar informações adicionais (itens de informação) e elementos complementares (elementos instrucionais – exemplos, exercícios, avaliações, entre outros) aos conceitos, denominada Modelo Instrucional. A terceira perspectiva tem o objetivo de estabelecer o sequenciamento de apresentação do conteúdo modelado, denominado Modelo Didático. A seguir esses modelos são detalhados e discutidos.

Figura 2 – AIM-CID: estrutura geral.



Fonte: adaptado (BARBOSA; BORGES; MALDONADO, 2013)

Modelo conceitual

O Modelo Conceitual tem como proposta descrever o domínio do conhecimento considerando uma visão de alto-nível. Em linhas gerais, uma instância do Modelo Conceitual define os conceitos relevantes para o entendimento de um determinado domínio. Além disso, relacionamentos entre conceitos são estabelecidos, formando, uma representação hierárquica do conhecimento. Para estruturação hierárquica dos conceitos, os relacionamentos podem ser estabelecidos considerando as seguintes características:

- **Relacionamentos Estruturais:** indicam se um conceito é de composição, quando um conceito é parte de outro (*part-of*), ou ainda, se um conceito é de classificação (*type-of*), isto é, quando um conceito integra outro mais genérico.
- **Relacionamentos Específicos do Domínio:** são relacionamentos com semântica própria de um determinado domínio. Assim, seu uso limita-se ao domínio de referência.

Para a modelagem dos conceitos do domínio, a AIM-CID propõe o uso da técnica de construção de Mapas Conceituais (NOVAK, 1990). A motivação de sua escolha é o amplo conhecimento da técnica pelos profissionais da educação e por sua facilidade de compreensão e uso. Além disso, por meio de Mapas Conceituais é possível representar os conceitos relevantes e a hierarquia estrutural do domínio do conhecimento (BARBOSA; BORGES; MALDONADO, 2013). A Figura 3 apresenta os passos para a construção do Modelo Conceitual considerando a inferência de relacionamentos entre os conceitos mais relevantes do domínio do conhecimento (BARBOSA; BORGES; MALDONADO, 2013).

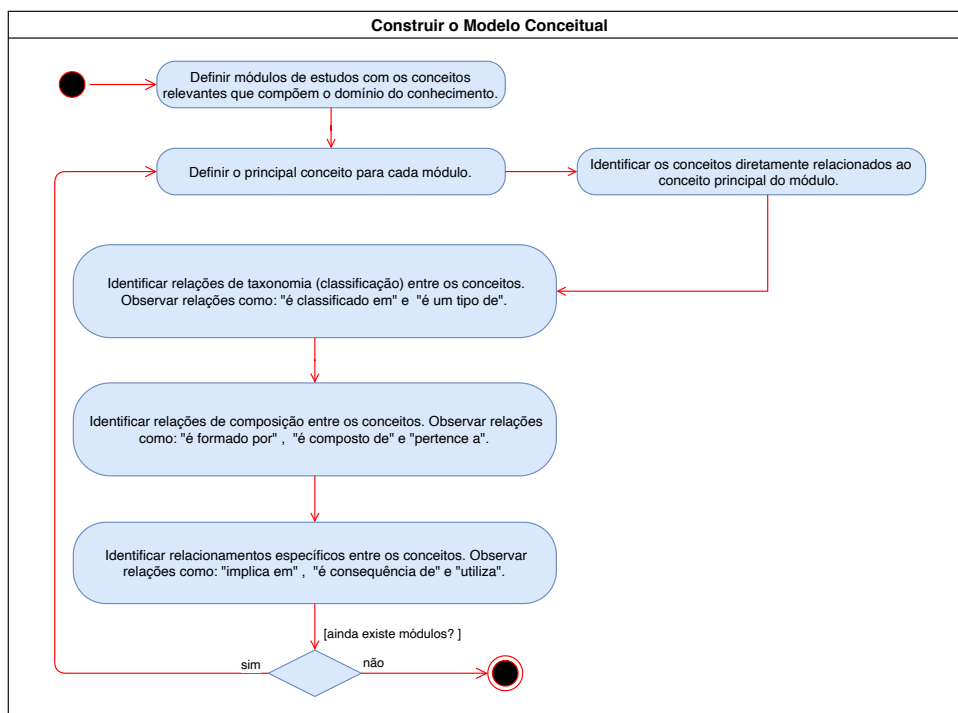
Em linhas gerais, a construção do Modelo Conceitual envolve segmentar o conteúdo educacional em módulos de aprendizado, definindo para cada módulo, os conceitos relevantes com base em seus inter-relacionamentos. Esse processo de identificação de conceitos e relacionamentos prossegue para todos os módulos propostos para o domínio do conhecimento modelado.

Modelo instrucional

Além da estrutura hierárquica de conceitos definidas com o Modelo Conceitual, um domínio de conhecimento apresenta um conjunto de informações adicionais que ajudam na sua compreensão. Na AIM-CID, essas informações adicionais são modeladas com a construção do Modelo Instrucional, que tem o objetivo de definir elementos e informações que ajudam o aprendiz assimilar cada conceito definido previamente. Logo, a construção de uma instância do Modelo Instrucional envolve as etapas de refinamento do Modelo Conceitual e definição de elementos instrucionais.

Na etapa do refinamento do Modelo Conceitual, considerando os conceitos definidos anteriormente, informações adicionais relacionadas ao domínio de conhecimento são atribuídas

Figura 3 – Passos para construção do Modelo Conceitual.



Fonte: elaborado pelo autor.

ao Modelo Instrucional com uso de *itens de informação*. *Itens de informação* representam elementos adicionais que permitem a compreensão de um dado conceito. A AIM-CID propõe o uso da teoria CDT (*Component Display Theory*) (MERRILL, 1983) para categorizar os *itens de informação*, considerado os seguintes tipos:

- **conceito**, representa entidades, símbolos, eventos e objetos que compartilham características e pode ser identificado por um nome único.
- **fato**, evento de cuja ocorrência se tem conhecimento, ou coisa cuja existência é inquestionável. Nomes, datas e eventos são exemplos de fatos.
- **procedimentos**, conjunto ordenado de passos, que visam a resolver um problema ou atingir um objetivo.
- **princípio**, explicações e deduções sobre o porquê de determinados acontecimentos e das maneiras específicas com que os mesmos ocorrem. Por exemplo, na Lógica tem-se o seguinte princípio:

“Princípio do terceiro excluído é aquele que rejeita a existência de qualquer meio-termo entre dois juízos contrários entre si, de tal maneira que um é necessariamente verdadeiro e o outro falso (MICHAELIS, 2020)”.

Na etapa de definição de *elementos instrucionais*, informações adicionais são utilizadas para complementar os *itens de informações* definidos previamente. *Elementos instrucionais* são informações que têm a finalidade de complementar um dado *item de informação* de maneira a melhorar e reforçar a sua compreensão. Os *elementos instrucionais* são categorizados como:

- **Elementos explanatórios**, que são informações que explicam um dado item de informação (exemplo, dica, sugestão de estudo);
- **Elementos exploratórios**, permitem o aprendiz explorar conceitos por intermédio de exercícios, simulações, ferramentas;
- **Elementos de avaliação**, responsáveis por medir a evolução do aprendizado.

Observando os *itens de informação* e os *elementos instrucionais* que compõe o Modelo Instrucional, pode-se inferir que o primeiro representa partes essenciais do conteúdo educacional e visa apresentar ao aprendiz os conceitos do domínio, enquanto o segundo representa parte do conteúdo educacional que é opcional dependendo do contexto didático e do objetivo da aprendizagem.

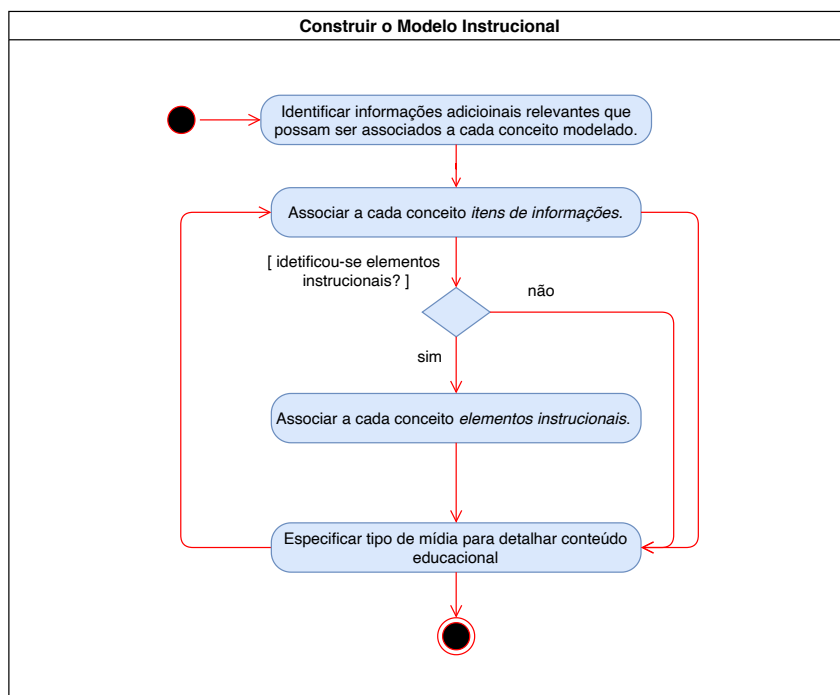
Como ilustra a [Figura 2](#) é no Modelo Instrucional que são atribuídos mídias como texto, imagem, vídeo, áudio, entre outros para detalhar cada *item de informação* e *elemento instrucional*. Por propor uma modelagem genérica de conteúdo educacional, a AIM-CID não sugere nenhum tratamento específico para elaboração de elementos de mídia considerando o contexto de ensino. No entanto, um conteúdo educacional preparado considerando as características de contexto, meio de apresentação (quadro, slide, app, web) e público-alvo facilita o processo de ensino e aprendizagem. Nesse sentido, o trabalho de pesquisa desenvolvido nesse doutorado propõe um guia de produção de conteúdo para aplicações educacionais móveis, que será apresentado no [Capítulo 3](#). A [Figura 4](#) apresenta os passos para construção do Modelo Instrucional ([BARBOSA; BORGES; MALDONADO, 2013](#)).

Modelo didático

O Modelo Didático tem o objetivo de organizar o sequenciamento de apresentação dos objetos caracterizados no Modelo Instrucional. Na prática, o Modelo Didático estabelece visões dos elementos modelados no Modelo Instrucional considerando cenários de ensino e aprendizagem. Basicamente, o Modelo Didático estabelece relações de precedências e relacionamentos didáticos para construção de sequências de apresentação do conteúdo educacional modelado no Modelo Instrucional (*itens de informação* e *elementos instrucionais*).

O sequenciamento do conteúdo educacional, bem como qual conteúdo é disponibilizado obedece à estratégia de ensino e aprendizagem implementada pelo professor. Além disso, é possível selecionar o conteúdo educacional que será utilizado em dado contexto de aprendizagem.

Figura 4 – Passo para construção do Modelo Instrucional.

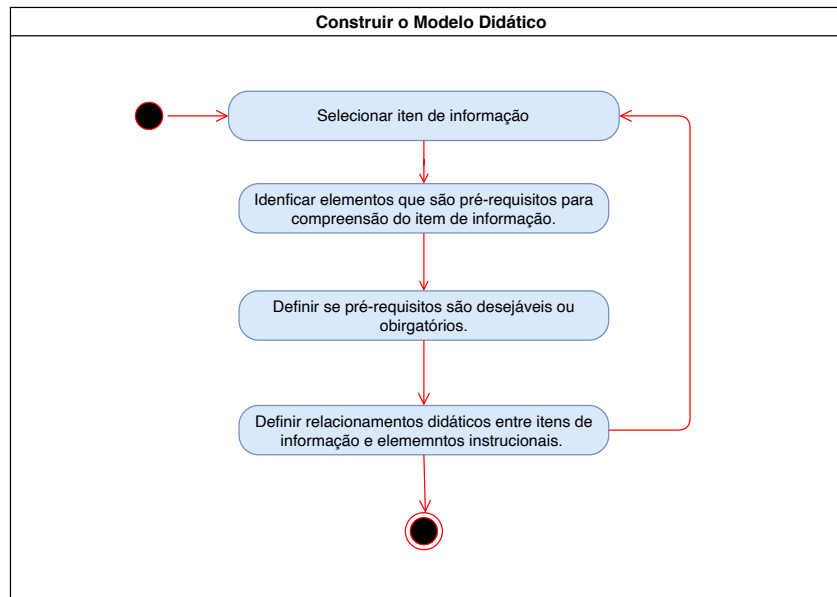


Fonte: elaborado pelo autor.

Uma possibilidade de sequenciamento de conteúdo é a especificação aberta. A especificação aberta permite que formas distintas de apresentação do conteúdo sejam representadas a partir de um mesmo modelo didático. Assim, o professor pode decidir a forma em que o conteúdo é apresentado e o aprendiz tem liberdade de navegar no conteúdo, sem restrições de acesso. A relação de precedência definida no Modelo Didático também permite um sequenciamento restritivo, construindo assim um caminho de navegação obrigatório no conteúdo. O Modelo Didático não impõe um formato de sequenciamento do conteúdo, o professor tem a liberdade de modelar conforme seus objetivos de ensino e aprendizagem.

A AIM-CID, por meio dos modelos Conceitual, Instrucional e Didático, permite uma estruturação genérica do conteúdo educacional de forma sistemática e modular. A separação de conteúdo educacional em visões distintas (conceitual, instrucional e didática) amplia a aplicação da abordagem. O Modelo Conceitual do domínio do conhecimento pode ser complementado com níveis diferentes de aprofundamento (Modelo Instrucional para diferentes audiências) e com formas variadas de apresentação (Modelo Didático com navegação flexível). Assim, a abordagem também pode ser estendida para atender requisitos de modalidades de ensino e aprendizagem emergentes.

Figura 5 – Passos para construção do Modelo Didático.



Fonte: Elaborada pelo autor.

2.3 Engenharia Dirigida por Modelos (*Model-driven engineering* – MDE)

O avanço tecnológico trouxe a necessidade que produtos tivessem em sua composição, de alguma forma, computadores e software. O software, então, torna-se essencial em atividades como controle de voo, indústria automobilística, sistemas bancários, entre outros. Com isso, surge também o aumento da complexidade e da necessidade de qualidade no desenvolvimento de software, pois falhas no desenvolvimento e execução do software podem trazer prejuízos financeiros (SOMMERVILLE, 2011). Assim, desenvolver software tornou-se mais complexo, exigindo qualidade e otimização de tempo de desenvolvimento.

Para atender a essa nova realidade uma série de técnicas ou paradigmas de desenvolvimento foram propostos: modelo em cascata, desenvolvimento evolucionário, transformação formal e reuso (SOMMERVILLE, 2011). Buscando atender a qualidade de desenvolvimento de software surge a abordagem da MDE (*Model-driven Engineering*). A MDE é uma proposta com o objetivo de gerenciar a complexidade de desenvolvimento de software, trazendo o modelo como foco principal na construção de um software (LOPES; HAMMOUDI; ABDELOUAHAB, 2005). Assim, o objetivo da MDE é elevar o nível de abstração usando modelos como entidades de primeira classe e, conseqüentemente, os artefatos primários de desenvolvimento (Lafi; Feki; Hammoudi, 2013). Um modelo representa situações do mundo real, além de permitir obter visões diferentes de um mesmo sistema (BÉZIVIN, 2006).

Para que modelos possam ser utilizados no processo de desenvolvimento de software, eles precisam ser concebidos em uma linguagem de modelagem bem definida, com sintaxe

e semântica que permitam expressar seus elementos e relações (LOPES, 2007). No contexto MDE, modelos são criados conforme um metamodelo. Um metamodelo especifica elementos que compõem um modelo, como estes elementos se relacionam e quais são as restrições impostas pelo modelo (BÉZIVIN, 2006). A seguir, duas propostas de abordagem de desenvolvimento baseado na MDE serão descritas.

2.3.1 Model driven architecture (MDA)

A Arquitetura Dirigida por Modelos (*Model-Driven Architecture – MDA*) é uma abordagem MDE proposta pela *Object Management Group* (OMG). Na abordagem MDA, o ciclo de vida de desenvolvimento do software segue o ciclo de vida de desenvolvimento de software tradicional, porém sua diferença está na criação de artefatos, que são modelos formais que podem ser entendidos e processados pelo computador (KLEPPE; WARMER; BAST, 2003). A OMG propõe a divisão da modelagem em quatro níveis de camada, descritas a seguir (Figura 6) (KLEPPE; WARMER; BAST, 2003):

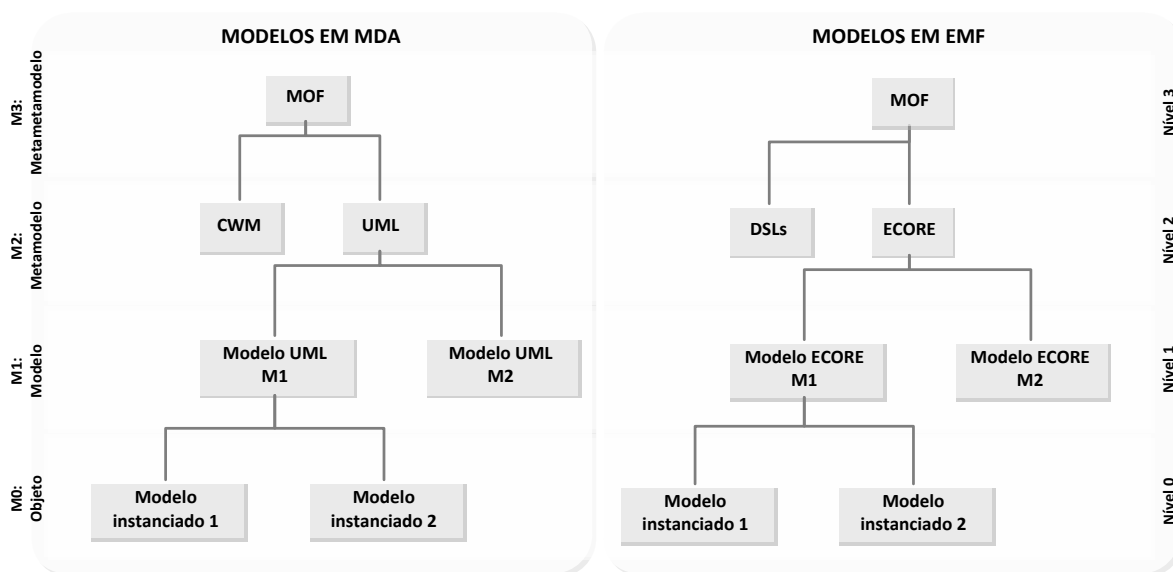


Figura 6 – Arquitetura de quatro camadas (BASSANI, 2012)

- **M0 (objeto):** camada que representa objetos do mundo real, isto é, são instâncias do domínio computacional de uma plataforma final. Como exemplo de objeto desta camada pode-se citar um aluno de um sistema acadêmico, onde este tem nome = "João Batista" e curso = "Ciência da Computação";
- **M1 (modelo):** camada que representa modelos de conceitos do mundo real. A camada M0 é formada por instâncias que estão conforme a camada M1, isto é, se M1 especifica que uma classe aluno é composta por nome e curso, as instâncias de M0 terão objetos alunos com essas propriedades;

- **M2 (metamodelo):** modelos criados nesta camada são denominados metamodelos. Um metamodelo especifica como um determinado modelo deve ser criado, especificando quais elementos e relações poderão ser criadas no modelo. Modelos da camada M1 são instâncias que estão conforme especificação da camada M2. Como exemplo, pode-se citar o metamodelo UML (OMG, 2010);
- **M3 (metametamodelo):** nesta camada, os conceitos necessários para criação de metamodelos são definidos. Um metamodelo deve ser criado conforme o que é especificado em seu metametamodelo.

A Figura 6 também apresenta as camadas do EMF (*Eclipse Modeling Framework*). As camadas do EMF possuem as mesmas funções das camadas propostas pela OMG. No EMF, o nível 3 representa os metametamodelos, no nível 2 tem-se a definição dos metamodelos conforme metametamodelos do nível 3, no nível 1 têm-se os modelos instanciados conforme metamodelos definidos no nível 2 e, por fim, tem-se o nível 0 com objetos instanciados conforme modelos do nível 1.

Para entender o processo de desenvolvimento no contexto MDA, alguns conceitos importantes devem ser conhecidos (Figura 7). Dentre eles destacam-se:

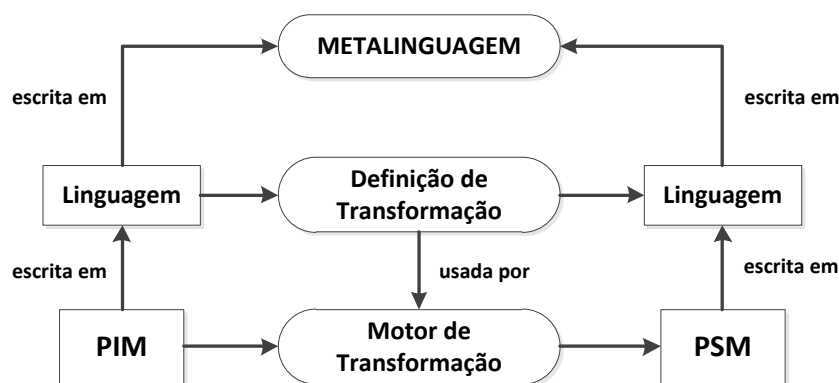


Figura 7 – Framework básico MDA (BASSANI, 2012)

- **Modelo Independente de Plataforma (PIM):** nesse modelo o projetista cria uma abstração de alto nível do problema, sem se ater a qualquer aspecto tecnológico de implementação da solução. No PIM, o sistema é modelado em uma visão de melhor solução para o negócio;
- **Modelo Específico de Plataforma (PSM):** ao definir esse modelo, o projetista descreve o sistema considerando os aspectos tecnológicos de sua implementação. O PSM é obtido a partir da execução de definições de transformação do modelo PIM;

- Definição de Transformação: permite definir como elementos de um modelo fonte serão transformados em elementos de um modelo alvo. Com a definição de transformação criada, pode-se então transformar qualquer modelo fonte em um modelo alvo;
- Motor de Transformação: definições de transformação são especificações que declaram como gerar um modelo a partir de outro. Para que a transformação se realize de fato é preciso que as definições de transformação sejam executadas por um motor de transformação, que recebe um modelo de entrada, executa as definições de transformação e gera como saída outro modelo.

O processo de transformação de modelos é essencial para a abordagem MDA. Como mostra a Figura 8, um código é obtido após a execução de sucessivas definições de transformação. Isto permite que alterações no modelo PIM sejam refletidas no código, aplicando novamente as definições de transformação.

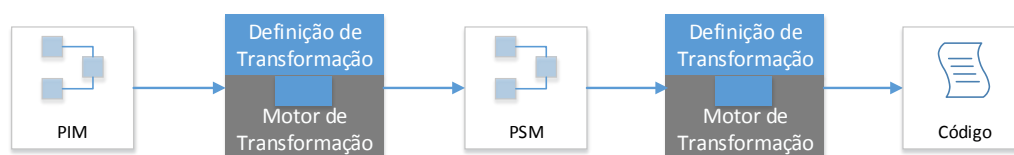


Figura 8 – Processo de transformação na abordagem MDA (KLEPPE; WARMER; BAST, 2003)

A MDA propõe-se a fornecer um melhor gerenciamento dos problemas de produtividade, portabilidade e interoperabilidade presentes no desenvolvimento de software (KLEPPE; WARMER; BAST, 2003), como apresentado a seguir:

- Produtividade: na abordagem MDA, os desenvolvedores concentram-se no trabalho de criação do modelo independente de plataforma, deixando aspectos da plataforma específica para a definição de transformação. Além disso, há menos código a ser escrito ao nível de PSM e código, já que grande parte dessa implementação é gerada pela transformação do PIM para PSM e do PSM para código;
- Portabilidade: um mesmo PIM pode ser transformado para plataformas diferentes;
- Interoperabilidades: a interoperabilidade é atingida com a criação de pontes entre PSM gerados a partir de um mesmo PIM. As pontes permitem que conceitos de uma plataforma sejam transformados em conceitos de outra plataforma.

A ideia básica da MDA é gerenciar o ciclo de desenvolvimento de software centrado na modelagem. Assim, a manutenção e evolução do software fica em alto nível de abstração, isto é, na criação e manutenção de modelos. A abordagem dirigida por modelo utilizada nesse

trabalho é implementada com o *Eclipse Modeling Framework* (EMF). O EMF é um *framework* que apoia a modelagem e geração de código na construção de ferramentas ou aplicações tendo como base uma estrutura de modelo de dados (STEINBERG *et al.*, 2008). O EMF será aplicado na construção dos metamodelos, na criação e visualização de modelos e na transformação de modelos.

2.4 Modelagem de conteúdo educacional: um mapeamento sistemático

A qualidade do conteúdo educacional utilizado em um processo de ensino e aprendizagem é um fator relevante para efetiva aprendizagem do estudante (BARBOSA; BORGES; MALDONADO, 2013). Nesse sentido, a tarefa de modelagem do conteúdo educacional é uma atividade que merece atenção na produção de material didático, pois define como conceitos são estruturados para atender os objetivos do processo de ensino e aprendizagem (BARBOSA; MALDONADO, 2011). Considerando esse contexto, um mapeamento sistemático foi conduzido com o objetivo de identificar trabalhos que investiguem abordagens de modelagem de conteúdo educacional para modalidades de ensino e aprendizagem baseadas em TICs. Optou-se por realizar um mapeamento sistemático que abrangesse não apenas a modalidade *m-learning*, com intuito de levantar o quanto de atenção a modelagem de conteúdo educacional tem recebido dos pesquisadores.

2.4.1 Metodologia, planejamento e execução

O mapeamento sistemático conduzido para identificar as principais propostas de modelagem de conteúdo educacional existentes na literatura seguiu as diretrizes estabelecidas por Kitchenham e Charters (2007). Quatro questões de pesquisa foram definidas para guiar a seleção dos trabalhos:

- Q1 Quais são as abordagens de modelagem de conteúdo educacional, encontradas na literatura, para as modalidades de ensino e aprendizagem *e-learning*, *m-learning* e *t-learning*?
- Q2 Quais são os métodos de avaliação que estão sendo utilizados para validar as abordagens de modelagem identificadas na questão Q1?
- Q3 Existem abordagens que forneçam apoio à produção de conteúdo educacional de forma automática ou semi-automática?
- Q4 Quais tecnologias são utilizadas nas abordagens levantadas na questão Q3?

Com base nas questões de pesquisa, termos de busca foram definidos para serem aplicados nas bibliotecas digitais e bases indexadas como: IEEE *Xplore* (ieeexplore.ieee.org), Scopus

(scopus.com), ACM *Digital Library* (dl.acm.org) e Compendex (www.engineeringvillage2.org). Foram, ainda, realizadas buscas na Revista Brasileira de Informática na Educação (RBIE) e nos anais dos seguintes congressos da área de Informática na Educação: Simpósio Brasileiro de Informática na Educação (SBIE), *Workshop* de Informática na Escola (WIE), *Workshop* sobre Educação em Computação (WEI).

As seguintes palavras-chave foram definidas para compor a *string* de busca: *learning*, *educational content*, *m-learning*, *mobile learning*, *e-learning*, *t-learning*, *content modeling*. Como as bibliotecas digitais e bases indexadas utilizam estrutura de construção de termos de busca diferentes, foi necessária a adaptação da *string* de busca, visando assim obter uma quantidade mais significativa de trabalhos. Para conduzir o processo de seleção de estudos primários foram definidos critérios de inclusão e exclusão., com o objetivo de filtrar os trabalhos coerentes com as questões de pesquisa definidas para o mapeamento. Além disso, foram descartados os trabalhos considerados resumidos ou incompletos. A Tabela 1 apresentam os critérios de inclusão e exclusão para seleção dos trabalhos:

Tabela 1 – Critérios de inclusão e exclusão de publicações.

Critérios de seleção		
#	Inclusão	Exclusão
1	Trabalhos que tratem de modelagem de conteúdo para as modalidades de ensino e aprendizagem.	Trabalhos considerados resumidos ou incompletos.
2	Em caso de estudos repetidos ou duplicados, quando houver trabalho com pesquisa mais recente, o trabalho mais recente é o selecionado.	Estudos duplicados.
3	Artigos completos em português ou inglês	Publicações com mais de 10 anos.

Fonte: Elaborada pelo autor.

2.4.2 Processo de seleção dos estudos primários

O processo seleção dos estudos primários consistiu na aplicação das *strings* de busca nas respectivas bases de estudos. Aos trabalhos recuperados foram aplicados os critérios de seleção de inclusão para identificar os estudos considerados relevantes. Nesta etapa a seleção dos estudos foi baseada na leitura de título, resumo e palavras-chave.

Após a seleção preliminar, a seleção final envolveu a etapa de exclusão de estudos com base na leitura dinâmica dos trabalhos e na construção de documento de extração de dados dos estudos incluídos. A execução do mapeamento aconteceu no período de agosto/2016 a outubro/2016. O resultado do mapeamento é sintetizado pela Tabela 2.

Tabela 2 – Número de trabalhos recuperados por base indexada.

#	BASE	TOTAL
1	IEEE	241
2	ACM	103
3	SCOPUS	760
4	COMPENDEX	225
5	SPRING LINK	193
6	SBIE	55
7	WIE	243
8	RBIE	86
Total		1906

Fonte: Elaborada pelo autor.

2.4.3 Seleção final e extração do resultados

Nesta etapa, inicialmente foram removidos 423 trabalhos duplicados dos 1.906 selecionados em todas as bases de busca, resultando assim em 1.483 trabalhos selecionados. A etapa seguinte, teve como objetivo realizar revisões para exclusão de trabalhos por leitura de títulos e resumos. Após, a atividade de leitura de títulos, 1.186 trabalhos foram descartados, restando 297 trabalhos. Dos 297 estudos, 193 foram excluídos, pela leitura de seus resumos. Portanto, o processo de seleção preliminar de trabalhos, obteve o total de 104 estudos.

Foram ainda excluídos 11 trabalhos repetidos, sendo selecionados os trabalhos mais recentes. Dezoito (18) trabalhos com mais de 10 anos de publicação foram excluídos do mapeamento. Assim, após o processo de seleção de trabalhos considerando os critérios de inclusão, 15 trabalhos foram recuperados para leitura completa.

2.4.4 Análise dos trabalhos selecionados

Nesta seção são sumarizados os estudos selecionados, os quais abordam questões de modelagem de conteúdo educacional. Os trabalhos selecionados são analisados considerando as questões de pesquisa definidas na Seção 2.4.1. A Tabela 3 apresenta os 15 estudos resultantes do processo de mapeamento.

Q1. Quais são as abordagens de modelagem de conteúdo educacional, encontradas na literatura, para as modalidades de ensino e aprendizagem *e-learning*, *m-learning*, *t-learning*?

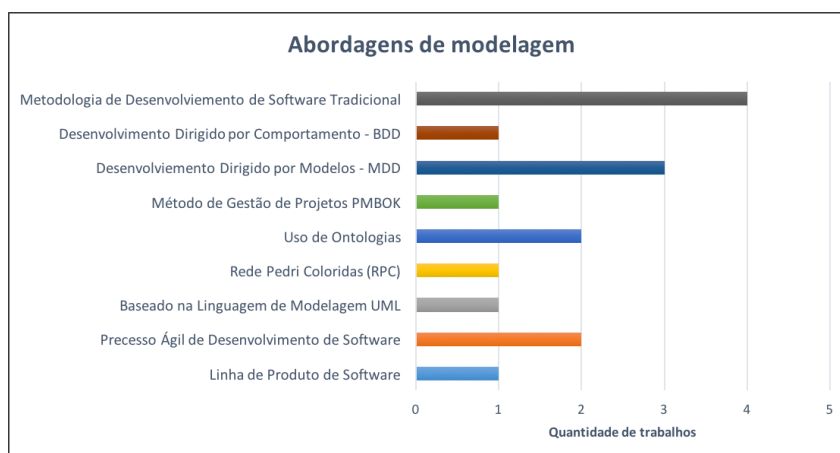
A Figura 9 apresenta as abordagens de modelagem de conteúdo que foram identificadas nos trabalhos estudados. Quatro trabalhos selecionados ([E09] [E11] [E14] [E15]) recorreram à abordagem de desenvolvimento de software tradicional. Foram identificados, ainda, traba-

Tabela 3 – Trabalhos selecionados

ID	Publicado em	Autor (Referência)	Título
E01	Proceedings of the 30th Annual ACM Symposium on Applied Computing	(LABIB <i>et al.</i> , 2015)	Enforcing reuse and customization in the development of learning objects: a product line approach
E02	IEEE Latin America Transactions	(GAMBOA; DURAN; BUILES, 2015)	MADCE - TVD - Model Agile Development Educational Content for Digital Television
E03	IEEE 12th International Conference on Emerging eLearning Technologies and Applications (ICETA)	(LANG; ČERVENÁK, 2014)	Use case driven educational content modeling with UML
E04	Anais do Simpósio Brasileiro de Informática na Educação	(Júnior <i>et al.</i> , 2014)	Um modelo para a produção de Objetos de Aprendizagem Acessíveis: Modelagem e Análise por Redes de Petri Coloridas
E05	Anais do Simpósio Brasileiro de Informática na Educação	(NOGUEIRA; AMORIM; LÓSCIO, 2014)	Desenvolvimento de uma ontologia para Recursos Educacionais Abertos e Fragmentados usando a abordagem METHONTOLOGY
E06	IEEE 13th International Conference on Advanced Learning Technologies	(CHIMALAKONDA; NORI, 2013)	IDont: An Ontology Based Educational Modeling Framework for Instructional Design
E07	Anais do Simpósio Brasileiro de Informática na Educação	(BRAGA; PIMENTEL; DOTTA, 2013)	Metodologia INTERA para o desenvolvimento de Objetos de Aprendizagem
E08	Anais do Simpósio Brasileiro de Informática na Educação	(SOUZA; FILHO; ANDRADE, 2012)	Customização Guiada: uma estratégia Orientada a Modelos para a Produção de Objetos de Aprendizagem
E09	Anais do Simpósio Brasileiro de Informática na Educação	(KEMCZINSKI <i>et al.</i> , 2012)	Metodologia para Construção de Objetos de Aprendizagem Interativos
E10	Frontiers in Education Conference (FIE)	(SILVA; BARBOSA; MALDONADO, 2011)	Model-driven development of learning objects
E11	Anais do Simpósio Brasileiro de Informática na Educação	(MORAES <i>et al.</i> , 2011)	Elaboração de Objetos de Aprendizagem para o LAPREN: Processo de Desenvolvimento e Sistema de Produção
E12	World Summit on Knowledge Society	(MINOVIĆ <i>et al.</i> , 2009)	Knowledge modeling for educational games
E13	Anais do Simpósio Brasileiro de Informática na Educação	(LAPOLL <i>et al.</i> , 2009)	Modelo de Desenvolvimento de Objetos de Aprendizagem Baseado em Metodologias Ágeis
E14	Revista, HÍFEN	(PESSOA; BENITTI, 2008)	Proposta de um Processo para Produção de Objetos de Aprendizagem
E15	Anais do Simpósio Brasileiro de Informática na Educação	(OLIVEIRA; NELSON; ISHITANI, 2007)	Ciclo de vida de objetos de aprendizagem baseado no padrão SCORM

Ihos propondo abordagens como o uso de Linha de Produto de Software [E01], Linguagem de Modelagem UML [E03], Redes Petri Coloridas (RPC) [E04], Ontologias [E05] [E06], Desenvolvimento Dirigido por Comportamento [E13] e Metodologia Ágil de Desenvolvimento

Figura 9 – Abordagens de modelagem utilizadas na produção de conteúdo educacional



Fonte: Elaborada pelo autor.

[E02] [E13]. Destacam-se ainda no mapeamento, a presença de três trabalhos com a proposta da aplicação da abordagem de Desenvolvimento Dirigido por Modelos (MDD) [E08][E10][E12] e um trabalho [07] baseado no guia PMBOK (DIONISIO, 2017). A seguir dois dos trabalhos com proposta de modelagem dirigida por modelos são apresentados.

Em [E08] é apresentada uma proposta de Customização Guiada para produzir Recursos Educacional Digitais (RED) que possam ser ajustados diretamente pelos professores. A proposta da Customização Guiada está apoiada na Engenharia de Software Orientada a Modelos (*Model-Driven Software Engineering* - MDSE). A Customização Guiada foi denominada de CLAssRoOM e consiste em três etapas: i) Descrição dos Requisitos, ii) Desenvolvimento de Componentes Gráficos ou simplesmente Modelagem e, iii) Implementação de Ações. Os requisitos definidos são utilizados para definição da arquitetura de Objetos de Aprendizagem Customizáveis (OAC), por uso de uma Linguagem Específica de Domínio (DSL). A arquitetura produzida pela DSL serve de base para gerar o Roteiro de Atividades a ser utilizado pela equipe de *design* e os *templates* de implementação, que serão utilizados posteriormente pela equipe de programação para adequações nos códigos gerados. Uma ferramenta de autoria visual foi proposta para a produção de OAC. O CLO Studio está dividido em Cenas (*Scenes*), Controles (*Controls*), Mídias (*Medias*) e Ações (*Actions*). O CLO Studio é uma ferramenta utilizada pela equipe de desenvolvimento para auxiliar a construção de OAC.

Em [E10] é apresentado um estudo que propõe a aplicação do Desenvolvimento Dirigido por Modelos (MDD) para implementação de OAs. O trabalho consiste na proposta do LODM (*Learning Object Development Method*) sendo uma extensão da abordagem IMA-CID (BARBOSA; MALDONADO, 2004). A abordagem IMA-CID (Seção 2.2) tem o objetivo de definir modelos Conceituais, Instrucionais e Didáticos para a modelagem de OAs. Para modelagem dos conceitos foram utilizados mapas conceituais, sendo esses representados por um perfil UML. Para as transformações de modelos foi desenvolvido a ferramenta LODE (*Learning Object Development Environment*). Com os modelos do conteúdo educacional definidos, regras de transformações são aplicadas para gerar OA, como, por exemplo *slides* no formato LaTeX.

Observou-se no resultado deste mapeamento sistemático que as propostas de modelagem estão centradas na modalidade de aprendizagem *e-learning*. Dos 15 trabalhos, apenas um apresentou proposta de abordagem de modelagem para modalidade *t-learning*. Como discutido na motivação dessa pesquisa, o movimento de adoção de dispositivos móveis como ferramenta de ensino e aprendizagem é crescente, sendo importante a condução de estudos que estruturam conteúdos educacionais para a modalidade *m-learning*. O trabalho desenvolvido no doutorado tem seu escopo na proposta de uma abordagem que atenda os requisitos de estruturação do conteúdo educacional para dispositivos móveis. Portanto, a abordagem EDUCOM.ML oferece a professores e desenvolvedores um guia que apoia o processo de produção do conteúdo para aplicações educacionais móveis. Nesse contexto, nenhum dos trabalhos levantados nesse mapeamento apresentou essa mesma proposta.

Q2. Quais são os métodos de avaliação que estão sendo utilizados para validar as abordagens de modelagem identificadas na questão Q1?

Estudo de caso e aplicação de questionários foram os métodos de avaliação utilizados com mais frequência para validar as propostas apresentadas nos trabalhos selecionados. Os trabalhos que não especificaram um método de avaliação foram categorizados como "não avaliado".

Q3. Existem abordagens que forneçam apoio a produção de conteúdo educacional de forma automática ou semi-automática?

De 15 trabalhos estudados, cinco apresentaram ferramentas para auxiliar a abordagem de modelagem de conteúdo educacional, como mostra a coluna "Mecanismo de apoio" da Tabela 4.

Tabela 4 – Resumo da respostas questões de pesquisa.

ID trabalho	Abordagem de modelagem	Métodos de avaliação	Mecanismo de apoio	Tecnologias de implementação
1	Linha de produto de software	Não avaliado	Sim	Web service
2	Processo ágil de desenvolvimento de software	Não avaliado	Não	
3	Linguagem de modelagem UML	Estudo de caso/questionário	Não	
4	Redes Petri Coloridas (RPC)	Estudo de caso	Não	
5	Uso de ontologias	Estudo de caso	Não	
6	Uso de ontologias	Não avaliado	Não	
7	Desenvolvimento de software tradicional	Estudo de caso	Não	
8	Desenvolvimento dirigido por modelos (MDD)	Estudo de caso/questionário	Sim	Eclipse Modeling Framework (EMF) / Java
9	Desenvolvimento de software tradicional	Não avaliado	Não	
10	Desenvolvimento dirigido por modelos (MDD)	Estudo de caso	Sim	Perfil UML / XML / CmapTools
11	Desenvolvimento de software tradicional	Estudo de caso	Sim	Não informado no trabalho
12	Desenvolvimento dirigido por modelos (MDD)	Estudo de caso	Sim	UML / XSLT Tools / Java
13	Processo ágil de desenvolvimento de software / BDD	Estudo de caso	Não	
14	Desenvolvimento de software tradicional	Estudo de caso	Não	
15	Desenvolvimento de software tradicional	Estudo de caso	Não	

Dos trabalhos selecionados, três (MINOVIĆ *et al.*, 2009) (SILVA; BARBOSA; MALDONADO, 2011) (SOUZA; FILHO; ANDRADE, 2012) aplicaram a abordagem Dirigida por Modelos no processo de modelagem. A principal característica desta abordagem é conduzir um desenvolvimento de software centrado na construção de modelos e na transformação destes modelos, de forma programável, em código-fonte ou outros modelos. Esta característica propicia mecanismos de auxílio a produção de conteúdo educacional. Os demais trabalhos implementaram ferramentas de apoio, utilizando-se de linguagens de programação para ambiente Web e *desktop*.

Novamente, *e-learning* foi a modalidade alvo dos trabalhos. Percebe-se, então, que as pesquisas precisam avançar no sentido de propor ferramentas para outras modalidades de ensino e aprendizagem. Além disso, o resultado de poucos trabalhos fornecendo mecanismos de apoio à produção de conteúdo educacional sugere que as pesquisas estão propondo melhorias no processo de modelagem, porém, ainda, é preciso avançar na proposta de mecanismos que facilitem o trabalho dos profissionais na produção de conteúdos educacionais.

Q4. Quais tecnologias são utilizadas nas abordagens levantadas na questão Q3?

Pode-se observar na Tabela 4 que cinco trabalhos implementaram mecanismos para facilitar o processo de produção de conteúdo educacional. O trabalho de Labib *et al.* (2015) recorreu à tecnologia *Web Service* para interligar os componentes da arquitetura LOAT (*Learning Object Authoring Tool*). Souza, Filho e Andrade (2012) implementaram a ferramenta *Customizable LeARning Objects Model-driven* (CLAssRoOM) com a abordagem proposta pelo Eclipse Foundation denominada *Eclipse Modeling Framework* (EMF). O EMF fornece um *framework* para construir metamodelos e definir regras de transformações de modelos.

A ferramenta LODE (*Learning Object Development Environment*) apresentada no trabalho de Silva, Barbosa e Maldonado (2011) usa a tecnologia de Perfil UML para implementação de um *Domain-Specific Language* (DSL) para definir modelos, além de utilizar a ferramenta Cmap-Tools para especificar mapa conceitual de domínio do conhecimento. O trabalho de (MORAES *et al.*, 2011) não apresentou as tecnologias utilizadas para construção da ferramenta de produção de OAs. Minović *et al.* (2009) utilizaram as tecnologias UML, linguagem de programação Java e XSLT *tools* para implementar a ferramenta de modelagem de conteúdo educacional para jogos.

Como pode ser observado nos estudos recuperados no MS, houve um predomínio da modalidade *e-learning* como o ambiente de aprendizagem para modelagem do conteúdo educacional. Além disso, a metodologia tradicional de desenvolvimento de aplicações foi a mais utilizada para implementar software educacional. No entanto, o mapeamento também apresentou alguns trabalhos adotando a Engenharia Dirigida por Modelos para desenvolvimento de soluções educacionais. A MDE foi a abordagem de desenvolvimento de aplicação escolhida para implementar a abordagem de modelagem proposta nessa pesquisa de doutorado. Finalmente, o mapeamento sugere uma carência de propostas de modelagem de conteúdo educacional para *m-learning* e desenvolvimento de aplicações educacionais móveis.

2.5 Considerações finais

As TICs e sua integração nas atividades sociais favorecem o surgimento de novas modalidades de ensino. Modalidades como *e-learning* e *m-learning* apresentam vantagens, mas também desafios. Um destes desafios é a estruturação do conteúdo educacional de forma sistemática e adequada para uso nestas modalidades, aproveitando assim suas vantagens e superando suas limitações.

No contexto da estruturação de conteúdo educacional, uma sistematização formal de modelagem adequada para modalidade *m-learning* torna-se uma atividade importante para o engajamento de professores e aprendizes no uso de dispositivos móveis no campo da educação. A abordagem AIM-CID é sistematiza e define etapas que estruturam o conteúdo educacional de forma genérica. Por isso, a incorporação na AIM-CID de elementos que aproveitem as vantagens e minimizem as limitações presentes na modalidade *m-learning* pode contribuir para sua adoção

por professores.

Além disso, as diferentes visões do conteúdo educacional proporcionadas pela modelagem AIM-CID favorecem a aplicação da MDE, que tem como artefato primário de desenvolvimento a construção de modelos e suas transformações em artefatos alvo. Com a aplicação da MDE pode-se definir um Modelo Conceitual que, por regras de transformação, geram o Modelo Instrucional e Didático para atender objetivos e público-alvo distintos. A estes modelos também é possível propor metamodelos que estruturam o conteúdo educacional modelado para modalidades de ensino e aprendizagem diversas. Assim, a partir de um único Modelo Conceitual pode-se gerar conteúdo educacional para ambientes formais como sala de aula (slides, pdf) e ambientes informais, como páginas web e aplicações educacionais móveis. Vale ressaltar que este trabalho de pesquisa tem como escopo a geração de conteúdo para aplicações educacionais móveis.

Este trabalho de pesquisa sugere a proposição de uma abordagem de modelagem de conteúdo educacional que combine a estruturação de conteúdo da AIM-CID com a MDE para oferecer a professores e desenvolvedores um guia para a produção de material didático adequado para aplicações educacionais móveis. No próximo capítulo, a abordagem de modelagem de conteúdos educacional é detalhada e discutida.

ABORDAGEM DE MODELAGEM DE CONTEÚDO EDUCACIONAL PARA M-LEARNING

Este capítulo discute o conjunto de metamodelos que apoiam a EDUCOM.ML – abordagem de Modelagem de Conteúdo Educacional para *m-learning* (**E**ducational **C**ontent **M**odeling **M**obile **L**earning). EDUCOM.ML é uma proposta de abordagem de modelagem que estende a AIM-CID com elementos para produção de conteúdo educacional para *m-learning*. Além disso, sua implementação é apoiada pela abordagem de desenvolvimento dirigida por modelos (MDE). A abordagem EDUCOM.ML visa guiar a equipe de produção de conteúdo e de desenvolvedores em um projeto de construção de aplicativo educacional para dispositivos móveis. Nesse sentido, a EDUCOM.ML especifica um conjunto de metamodelos que atendem tanto a estruturação do conteúdo educacional, bem como modelam a plataforma móvel de apresentação do conteúdo modelado.

O capítulo é organizado pela Seção 3.1, que contextualiza a EDUCOM.ML em relação a aplicação da abordagem MDE na modelagem de conteúdo educacional e implementação de aplicativo educacional móvel; a Seção 3.2 que apresenta uma visão geral da EDUCOM.ML que detalha a divisão de responsabilidade dos envolvidos na produção de conteúdo educacional para aplicativos educacionais móveis; a Seção 3.3, que detalha os metamodelos propostos para estruturação do conteúdo educacional; a Seção 3.4, que detalha os metamodelos que auxiliam a construção da aplicação móvel e, por fim, considerações finais são apresentadas na Seção 3.6.

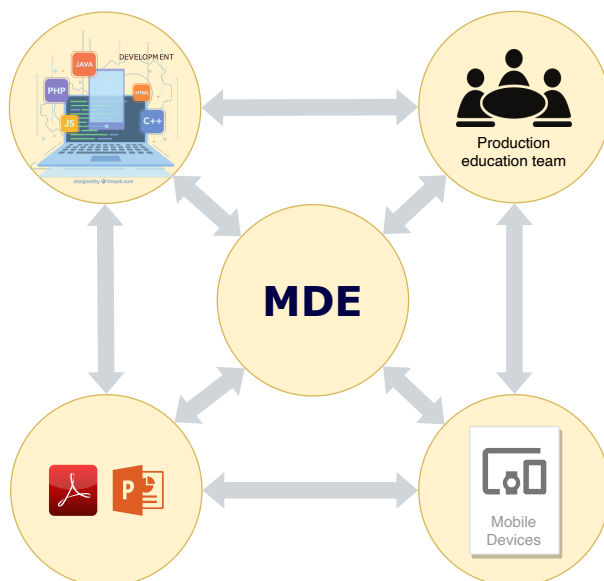
3.1 Visão Geral

Segundo (PRESSMAN; MAXIM, 2016) o processo de desenvolvimento de aplicações envolve o gerenciamento de interesses de diferentes grupos de pessoas (*stakeholders*). Em geral,

o desenvolvimento envolve usuários (especialistas no domínio) e desenvolvedores especializados na tecnologia utilizada para produção de aplicações. Esse contexto não difere quando o desenvolvimento é uma aplicação educacional, pois existe a equipe de produção de conteúdo educacional (professores/autores) e a equipe de desenvolvimento (programadores/analistas). A equipe de produção é composta por especialistas no domínio do conhecimento, isto é, profissionais que dominam todos os conceitos de uma área específica (física, matemática, engenharia de software, arquitetura, entre outras). No entanto, é comum que esses especialistas não dominem a tecnologia envolvida no processo de desenvolvimento de aplicações. Para que um software educacional alcance seu objetivo de proporcionar o aprendizado ao estudante, o conteúdo educacional precisa ser produzido com qualidade e considerando as restrições e recursos fornecidos por cada tecnologia adotada (BARBOSA; BORGES; MALDONADO, 2013).

Nesse contexto, a EDUCOM.ML tem a proposta de construir uma ponte entre a equipe de produção de conteúdo educacional e desenvolvedores. Essa ponte é apoiada pela abordagem da Engenharia Dirigida por Modelos (MDE). Segundo (IVANOV; BÉZIVIN; AKSIT, 2002) a MDE é uma abordagem que fornece recursos para construção de pontes entre espaços tecnológicos (domínios do conhecimento). Os espaços tecnológicos não são isolados, portanto, pontes são necessárias para que operações que são melhores desempenhadas em uma tecnologia, sejam aproveitadas por outras (IVANOV; BÉZIVIN; AKSIT, 2002). Como ilustra a Figura 10, a MDE abstrai a complexidade de um espaço tecnológico e aproveita seu resultado em outro. Explorar essa capacidade de minimizar a complexidade tecnológica para professores é um objetivo da abordagem EDUCOM.ML. Nas seções seguintes, é detalhada como a EDUCOM.ML propõe guiar professores na produção do conteúdo educacional e apoiar os desenvolvedores na construção de aplicações educacionais móveis.

Figura 10 – Pontes entre espaços tecnológicos.



Fonte: adaptado de (IVANOV; BÉZIVIN; AKSIT, 2002).

3.2 Abordagem Educom.ml

No contexto da estruturação do conteúdo educacional para dispositivos móveis, a EDUCOM.ML considera dois espaços tecnológicos. O primeiro espaço tecnológico tem a preocupação de estabelecer uma visão da modelagem do conteúdo educacional desconsiderando aspectos de implementação da aplicação móvel. Isso significa, por exemplo, que não é uma preocupação da equipe de produção de conteúdo conhecer características técnicas da linguagem de desenvolvimento utilizada na implementação da aplicação móvel. Contudo, é recomendável que a equipe de produção de conteúdo conheça aspectos de elaboração adequada de mídias (texto, imagem, vídeo), além da estruturação do conteúdo educacional que será utilizado na aplicação educacional móvel (CURUM; KHEDO, 2020). Nesse sentido, a EDUCOM.ML fornece um conjunto de diretrizes de produção de mídia, bem como um conjunto de modelos que guiam a produção do conteúdo educacional. Diferente do primeiro espaço tecnológico, o segundo espaço tecnológico considera as características técnicas na implementação do conteúdo educacional em dispositivos móveis. Portanto, características como linguagem de programação, base de dados e uso de recursos dos dispositivos móveis são de responsabilidade da equipe de desenvolvimento.

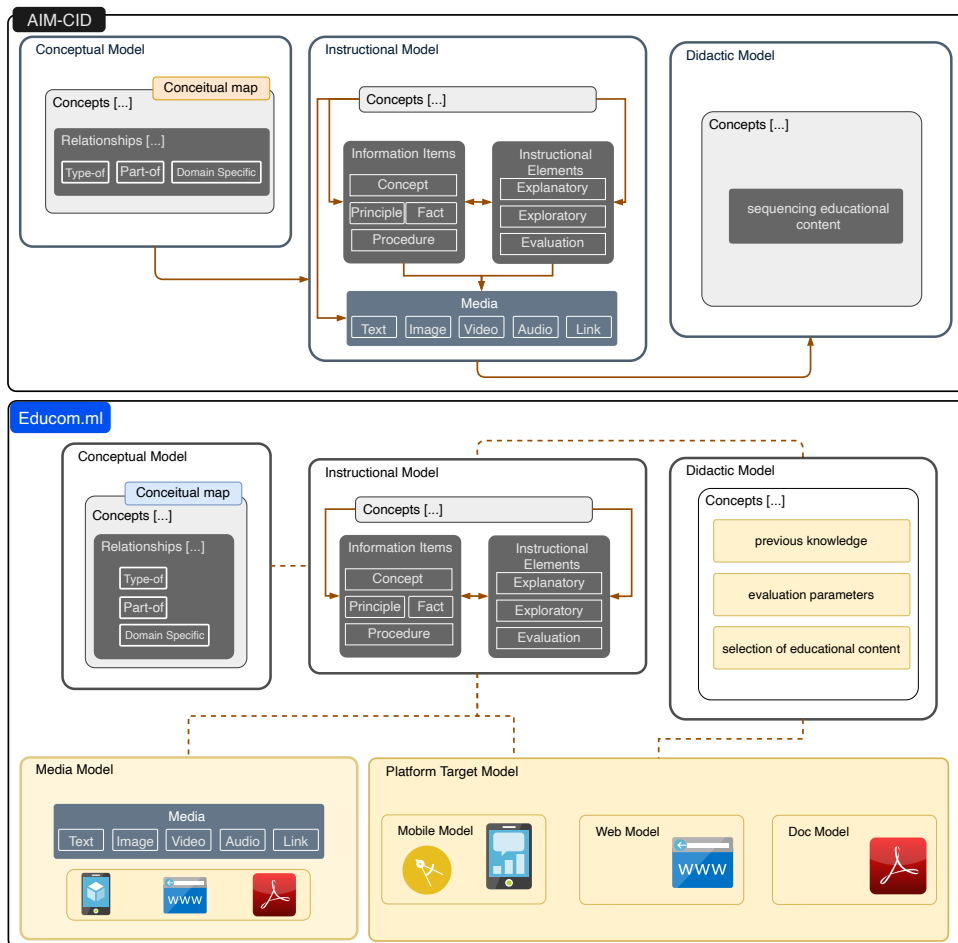
A Figura 11, apresenta uma visão geral das adequações realizadas na proposta original da abordagem AIM-CID para atender a produção de conteúdo educacional para *m-learning*. Em linhas gerais, foram mantidos os modelos propostos pela AIM-CID para estruturação do conteúdo educacional. No entanto, foram propostas alterações nos modelos Instrucional e Didático. Na Figura 11, os elementos em amarelo representam as alterações propostas. Analisando a figura, identifica-se que os elementos de mídia do Modelo Instrucional foram desmembrados, criando o modelo de mídia. Além disso, o Modelo Didático foi reestruturado, dando origem aos elementos conhecimento prévio, parâmetros de avaliação e seleção de conteúdo. Como elemento adicional, foi proposto um modelo para representar a plataforma alvo de implementação de aplicações educacionais móveis, o Modelo Mobile.

Ressalta-se que o escopo da EDUCOM.ML é a aprendizagem móvel, no entanto, é possível uma extensão para atender outras modalidades de ensino e aprendizagem. Como os modelos apresentados na EDUCOM.ML são implementados com a abordagem MDE, é possível fornecer uma arquitetura que possibilite a modelagem do conteúdo educacional para diferentes formatos. Essa extensão é obtida com a proposta de metamodelos que estruturam o conteúdo educacional para novas modalidades de aprendizagem. No atual momento da pesquisa esses metamodelos não foram propostos, porém, é uma atividade que está no planejamento de trabalhos futuros. Nas seções seguintes as adequações propostas pela EDUCOM.ML são apresentadas.

3.2.1 Modelo Conceitual da Educom.ml

O Modelo Conceitual foi mantido com os elementos da proposta original da AIM-CID, no entanto, foram enfatizadas algumas orientações para atender a plataforma de aprendizagem móvel.

Figura 11 – Proposta de extensão da abordagem AIM-CID.



Fonte: Elaborada pelo autor.

Devido à característica de mobilidade de dispositivos móveis no processo de aprendizagem, a sugestão de divisão do conteúdo educacional em módulos concisos de ensino foi enfatizada (DIWANJI *et al.*, 2014) (CHEN; LIN, 2016) (BIARD; COJEAN; JAMET, 2017) (MAYER, 2017).

Ao particionar o conteúdo em módulos menores, professores estão proporcionando ao aprendiz a possibilidade de otimizar seu tempo de estudo, isto é, qualquer intervalo de tempo livre pode ser um momento de aprendizado. Por exemplo, o aprendiz pode utilizar o tempo gasto no deslocamento entre casa e trabalho para estudar um módulo ou um conceito de um domínio de seu interesse. Além disso, o conteúdo ofertado em porções menores teoricamente consome menos tempo de estudo, dando ao aprendiz a sensação de evolução no conteúdo. No mais, o Modelo Conceitual manteve a característica original de estabelecer uma hierarquia de conceitos que se deseja ensinar.

3.2.2 Modelo Instrucional da Edcucom.ml

O Modelo Instrucional representa as informações que explicam os conceitos identificados para o domínio do conhecimento. Como ilustra a Figura 11, a EDUCOM.ML manteve a estruturação das informações da AIM-CID, isto é, os conceitos podem ser definidos com *Itens de Informação* e *Elementos Instrucionais*. Todavia, houve uma separação do design de mídia.

De fato, *Itens de Informação* e *Elementos Instrucionais* são materializados no conteúdo educacional por mídias (texto, áudio, vídeo e imagens). Contudo, em sua origem, a AIM-CID não propôs diretrizes ou qualquer orientação de produção de mídia que considerasse à plataforma de aprendizagem alvo. No contexto da aprendizagem móvel, estudos sugerem que produzir conteúdo educacional com atenção as limitações e características dos dispositivos móveis contribui para que o aprendiz compreenda os conceitos apresentados (WANG; SHEN, 2012) (PAULINS; BALINA; ARHIPOVA, 2015) (CHEN; LIN, 2016) (MAYER; FIORELLA; STULL, 2020).

Portanto, com o propósito de suprir a ausência de diretrizes de produção de mídia e para adequar o conteúdo educacional ao ambiente de aprendizagem móvel, a EDUCOM.ML propõe um modelo específico para a produção de mídias. As diretrizes sugeridas visam auxiliar produtores de conteúdo a produzir conteúdo educacional alinhado com as características de dispositivos móveis. A seguir as sugestões de design de mídia para dispositivos móveis são apresentadas.

3.2.3 Design de mídia para dispositivos móveis - Media Model

Considerando a modelagem de conteúdo para dispositivos móveis, a EDUCOM.ML apresenta um conjunto de recomendações para a produção de conteúdo educacional. Essas recomendações foram levantadas na literatura e junto a fabricantes de sistemas operacionais para dispositivos móveis (CHITTARO, 2010)(ELIAS, 2011)(DIWANJI *et al.*, 2014)(CHEN; LIN, 2016)(BIARD; COJEAN; JAMET, 2017) (APPLE, 2018)(GOOGLE, 2018)(MUTLU-BAYRAKTAR; COSGUN; ALTAN, 2019)(MAYER; FIORELLA; STULL, 2020). Foram identificadas diretrizes relacionadas com a apresentação de textos, imagens e vídeos. Além disso, as diretrizes foram classificadas em **didáticas [D]**, em que o professor é responsável em considerá-la, e em **técnicas [T]**, que merecem atenção da equipe de desenvolvimento. A seguir as diretrizes são apresentadas.

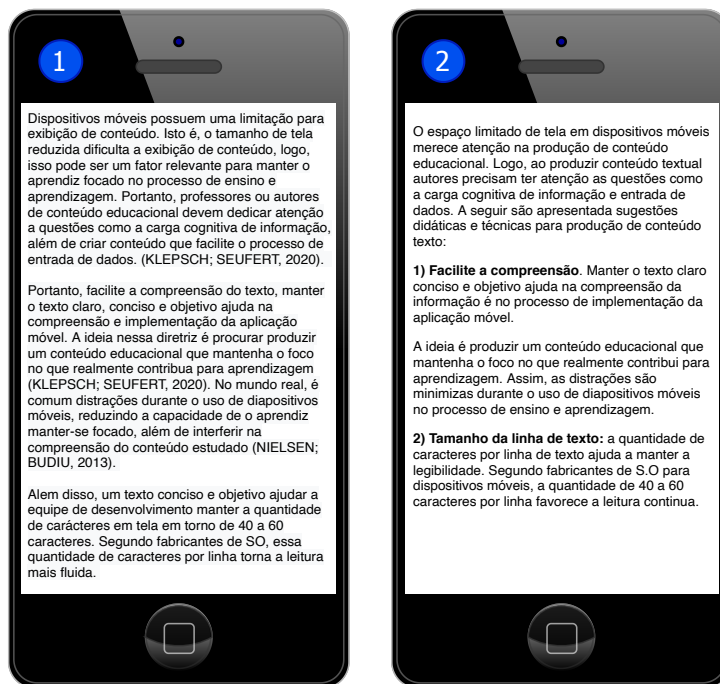
3.2.3.1 Texto

Dispositivos móveis apresentam espaço restrito para apresentação de conteúdo, merecendo assim, atenção quanto à elaboração de textos. Logo, ao produzir conteúdo textual os autores precisam considerar questões como a carga cognitiva de informação e o mecanismo de entrada de dados (CHITTARO, 2010)(ELIAS, 2011)(CHEN; LIN, 2016)(MUTLU-BAYRAKTAR; COSGUN; ALTAN, 2019)(KLEPSCH; SEUFERT, 2020). Como recomendações didáticas e

técnicas para produção de conteúdo textual destacam-se:

- I) **Facilite a compreensão [D]**. Manter o texto claro, conciso e objetivo ajuda na compreensão e implementação da aplicação móvel (CHITTARO, 2010)(NIELSEN; BUDIUI, 2013). A ideia nessa diretriz é produzir um conteúdo educacional que mantenha o foco no que realmente é relevante para aprendizagem, evitando a presença no texto de informações desnecessárias ao entendimento de um conceito (KLEPSCH; SEUFERT, 2020). Ao utilizar dispositivos móveis é comum que o aprendiz tenha distrações, reduzindo sua capacidade de manter-se focado, podendo assim interferir na compreensão do conteúdo estudado (NIELSEN; BUDIUI, 2013). A Figura 12 apresenta um exemplo do uso de um texto claro e objetivo. Pode-se observar que as interfaces “1” e “2” tratam sobre o mesmo tema, porém o conteúdo da interface “2” foi apresentado em uma estrutura mais objetiva e mantendo apenas as informações relevantes para a compreensão do texto.

Figura 12 – Texto conciso, claro e objetivo facilita a visibilidade do conteúdo.

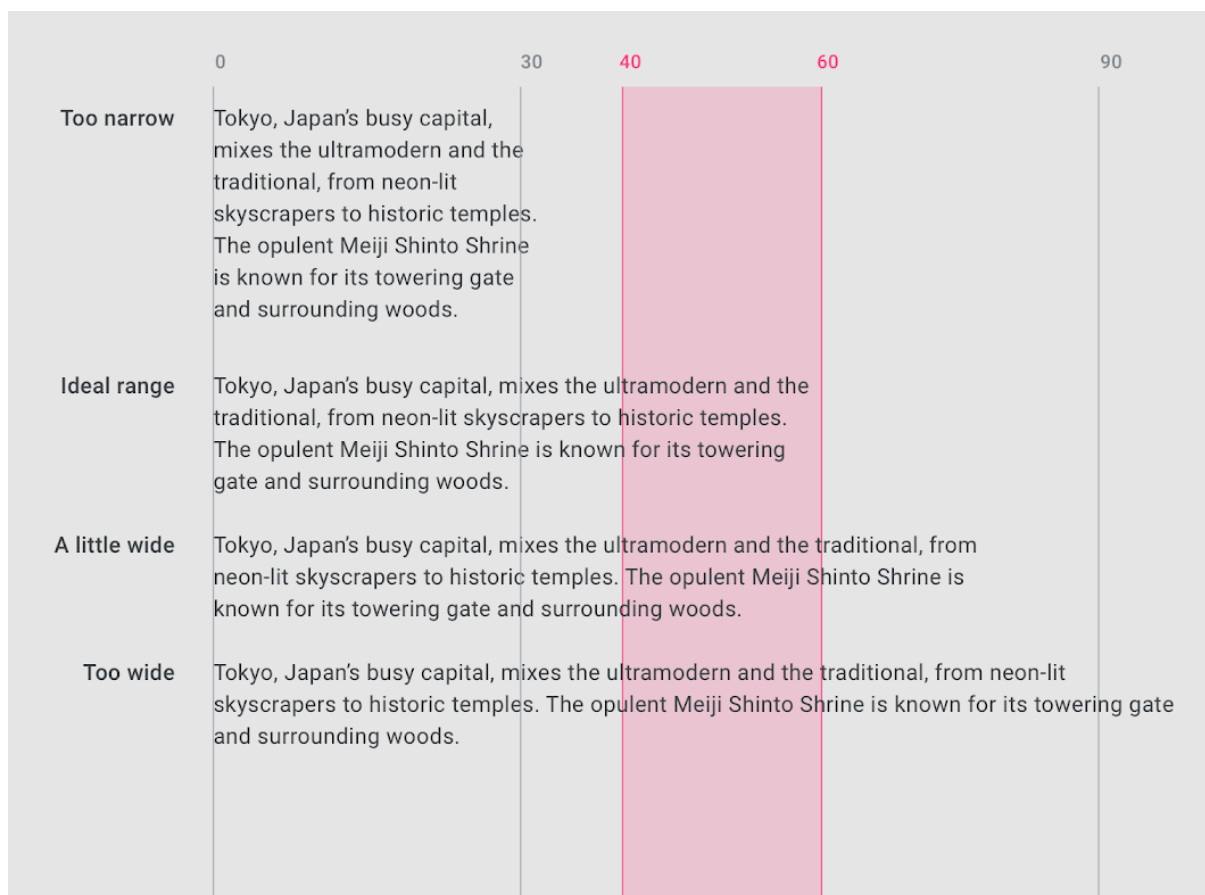


Fonte: Elaborada pelo autor.

- II) **Tamanho da linha de texto [D/T]**. A quantidade de caracteres por linha de texto é um parâmetro relacionado à legibilidade. Segundo fabricantes de Sistemas Operacionais para dispositivos móveis, linha de texto com 40 a 60 caracteres favorecem uma leitura mais contínua (GOOGLE, 2018). Textos longos que demandam mais tempo de leitura podem causar desconforto aos usuários (APPLE, 2018) (CHITTARO, 2010). Contudo, linhas de textos muito curtas prejudicam a legibilidade (GOOGLE, 2018). A implementação dessa diretriz envolve questões didáticas e técnicas. Produzir o conteúdo educacional conciso e

objetivo é uma prática didática que ajuda a equipe de desenvolvimento manter o padrão de tamanho de linha de texto sugerida. A Figura 13 ilustra a diferença de legibilidade considerando a quantidade de caracteres por linha. Observa-se que o texto com tamanho entre 40 e 60 caracteres oferece melhor legibilidade.

Figura 13 – Sugestão de tamanho de linha de texto.

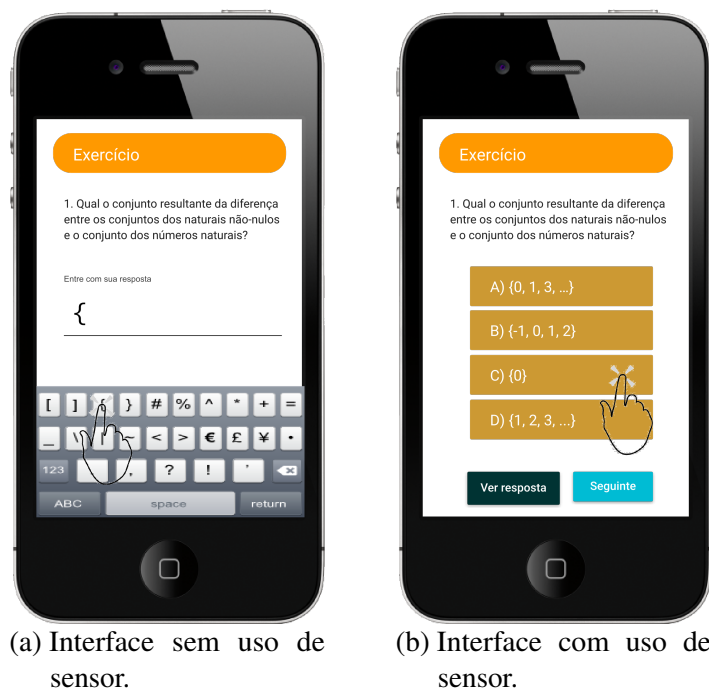


Fonte: (GOOGLE, 2018)

- III) **Escolha de fonte e cores [D/T]**. A variação de tamanho de fonte e cores é uma forma de enfatizar itens importantes no conteúdo apresentado em aplicativos móveis (APPLE, 2018). Portanto, professores podem identificar no texto os termos que precisam de destaque para que a equipe de desenvolvimento implemente este requisito.
- IV) **Controle de quebra de linha [T]**. Na quebra de linha, deve-se minimizar presença de grandes lacunas, preposições ou palavras curtas no final de linha (GOOGLE, 2018). Essa é uma prática que melhora a legibilidade do texto.
- V) **Legibilidade de texto sobre imagem [T/D]**. Para que a legibilidade de texto sobre imagens não seja prejudicada, aplique uma camada de gradiente. O gradiente deve ter uma opacidade que torne a legibilidade do texto natural e não prejudique a visibilidade da imagem (GOOGLE, 2018).

VI) **Utilize sensores dos dispositivos móveis [T/D]**. Uso dos sensores dos dispositivos móveis (captura de áudio, vídeo, GPS, câmera) pode auxiliar a entrada de dados (GOOGLE, 2018) (APPLE, 2018). A Figura 14 ilustra o uso do sensor de entrada em uma interface que implementa uma atividade avaliativa. Observa-se na Figura 14a que a interface de entrada de dados faz uso do teclado, tornando mais lento a interação do aprendiz com o aplicativo móvel. Na Figura 14b, a mesma atividade é apresentada ao aprendiz utilizando o recurso de toque de tela para entrada de dados. Fica nítido, que a segunda estratégia de entrada de dados é mais indicada para aplicativos educacionais móveis, pois facilita a interação do aprendiz com o aplicativo.

Figura 14 – Uso de sensores dos dispositivos móveis.



Fonte: Elaborada pelo autor.

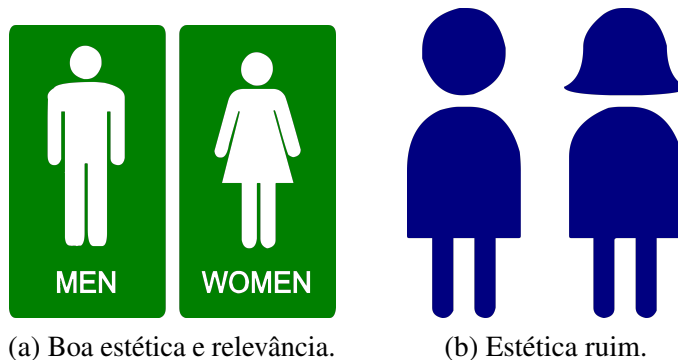
3.2.3.2 Imagem

Fotos e ilustrações caracterizam conceitos de maneira mais preditiva para os usuários de aplicações móveis (GOOGLE, 2018). Contudo, diretrizes de produção de imagem precisam ser seguidas para atingir a expressividade desejada. Nesse sentido, as seguintes diretrizes de *design* UI foram identificadas:

I) **Atenção a estética e relevância [D]**: utilize imagens que tornem a compreensão fácil e imediata (APPLE, 2018). É comum que o aprendiz esteja em deslocamento ou em um ambiente de ruído quando realiza a leitura de material didático em dispositivos móveis. Portanto, elaborar figuras que minimizam a carga cognitiva do aprendiz ajuda a compreensão do conteúdo (MAYER; FIORELLA; STULL, 2020). Observe a Figura 15 e veja como

a Figura 15a apresenta elementos mais relevantes em sua composição, oferecendo uma compreensão clara da informação do que a Figura 15b.

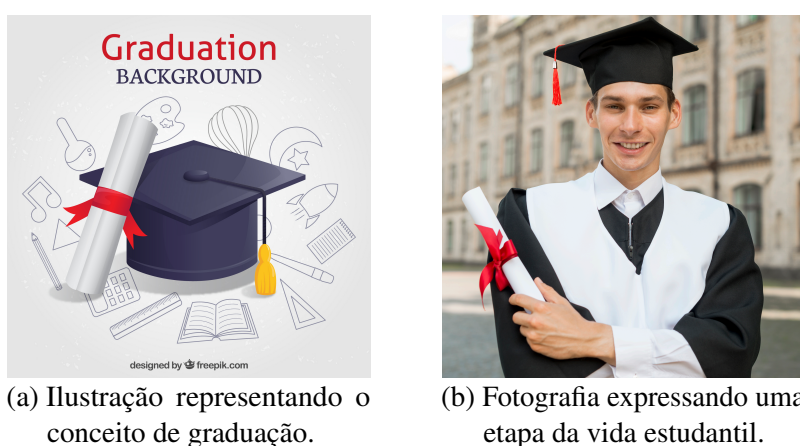
Figura 15 – Imagem com estética e relevância.



Fonte imagem: (COMPANY, 2020)

II) **Faça uso de ilustrações e fotografia [D]:** fotografias são melhor aproveitadas para expressar entidades e história específicas, enquanto ilustrações representam melhor conceitos e metáforas (APPLE, 2018). Observe como a ilustração apresentada na Figura 16a representa melhor o conceito de graduação, enquanto a fotografia da Figura 16b ilustra melhor um momento da vida acadêmica do aprendiz.

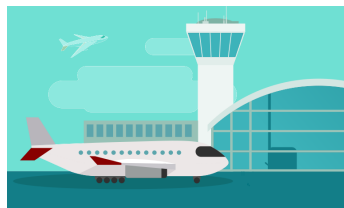
Figura 16 – Ilustração e fotografia.



Fonte imagem: (COMPANY, 2020)

III) **Mantenha um foco icônico na imagem [D/T]:** uma imagem com um foco claro expressa um conceito imediatamente (APPLE, 2018). O foco icônico ajuda a minimizar a carga cognitiva do aprendiz a medida que a informação presente na imagem é compreendida facilmente. Observe as Figuras 17a e 17b e veja como a primeira imagem apresenta com clareza a identificação do conceito de aeroporto do que a segunda imagem.

Figura 17 – Foco Icônico.



(a) Imagem com foco icônico



(b) Imagem foco não icônico.

Fonte imagem: (GOOGLE, 2018)

IV) **Crie imagens com construção narrativa [D]**. Imagens que narram uma história são mais informativas e compreensíveis (GOOGLE, 2018) [D]. Observe a Figura 18 e veja como é natural inferir que a imagem refere-se a questões comerciais entre duas nações.

Figura 18 – Imagem que conta uma história. Disputa comercial entre nações.



Imagem: Andrei Morais / Shutterstock

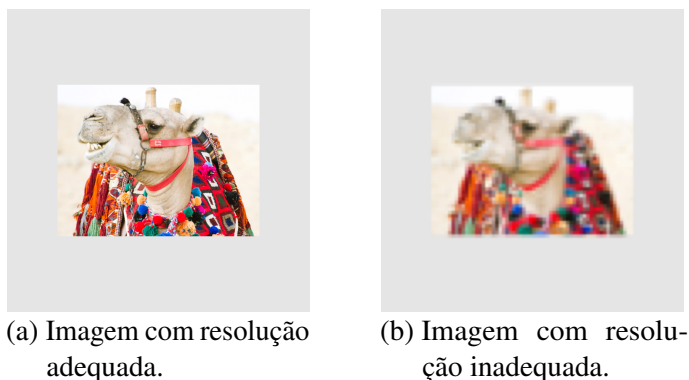
V) **Atenção a dimensões e resolução da imagem [D/T]**: imagens devem ser produzidas com resolução adequada ao procedimento de zoom, bem como as dimensões devem ser definidas considerando o tamanho da tela do dispositivo móvel (GOOGLE, 2018) (APPLE, 2018). Portanto, a resolução deve permitir a ação de zoom sem a degradação da imagem. Observe que a Figura 19b perde qualidade ao ser ampliada.

VI) **Utilize rótulos de texto [D/T]**. Atribua rótulo de texto a imagem para que o aprendiz possa se beneficiar do recurso de leitura de tela disponíveis em sistemas *android* e IOS (GOOGLE, 2018) (APPLE, 2018).

3.2.3.3 Vídeo

O consumo de informação por meio de vídeo é comum nos dias atuais. No campo da educação, vídeo também vem sendo uma ferramenta adotada no processo de ensino. O vídeo está presente tanto em plataformas web (Coursera, Alura, Udemy), como também em aplicações

Figura 19 – Resolução de imagem.



Fonte imagem: (COMPANY, 2020)

educacionais móveis. No entanto, a produção de vídeo educacional exige dos produtores de conteúdo alguns cuidados. Na intenção de auxiliar professores no trabalho de produção de vídeo algumas diretrizes são apresentadas a seguir:

- I) **Duração do vídeo [D]**. Conforme pesquisas e plataformas de visualização de vídeos, os usuários sentem-se mais confortáveis visualizando vídeos com duração entre 5 a 10 minutos (GOOGLE, 2018) (MADARIAGA *et al.*, 2021). Essa diretriz visa proporcionar ao aprendiz a possibilidade de estudo em momentos de deslocamento ou de tempo livre. Exemplificando, um vídeo de curta duração pode ser visualizados pelo aprendiz durante o deslocamento do trabalho para casa.
- II) **Segmente o conteúdo [D]**. A divisão do conteúdo em porções menores melhora a retenção do conhecimento (DIWANJI *et al.*, 2014) (BIARD; COJEAN; JAMET, 2017) (MAYER, 2017) (MYLLYMÄKI; HAKALA; HÄRMÄNMAA, 2018). Além disso, a segmentação contribui para que o aprendiz otimize o tempo de estudo. Assim, o conteúdo apresentado em vídeo de curta duração facilita seu estudo no tempo livre do aprendiz. Vale ressaltar que a segmentação precisa manter uma estrutura lógica de apresentação do conteúdo, permitindo assim que o aprendiz compreenda o domínio estudado (DIWANJI *et al.*, 2014) (BIARD; COJEAN; JAMET, 2017) (CASTRO-ALONSO *et al.*, 2021). Além disso, a segmentação contribui para o alinhamento com a diretriz de tempo de vídeo.
- III) **Conteúdo relevante ao curso [D]**. Vídeos que apresentam material didático diretamente relacionado ao conteúdo do programa do curso são mais visualizados (DIWANJI *et al.*, 2014). Portanto, ao produzir vídeos evite uso de recurso que não estão relacionados ao conteúdo do curso (MAYER; FIORELLA; STULL, 2020).
- IV) **Qualidade de edição do vídeo [D/T]**. Vídeos com uma aparência profissional, com recurso e visual de qualidade são mais envolventes (DIWANJI *et al.*, 2014) (MADARIAGA

et al., 2021). Qualidade de iluminação e de captação de áudio são exemplo de cuidados que tornam o vídeo com boa edição.

V) **Mantenha um discurso envolvente [D]**. Produza vídeo com diálogo mais leve e bem-humorado e mantenha uma velocidade do discurso que não cause monotonia no aprendiz (DIWANJI *et al.*, 2014).

A Tabela 5 apresenta uma visão geral das diretrizes para elaboração de mídia proposta pela EDUCOM.ML. A finalidade desse guia de design de mídia é orientar o professor na tarefa de elaborar conteúdo educacional considerando as restrições e os recursos de dispositivos móveis. Ao seguir essas orientações, professores facilitam o trabalho de implementação dos desenvolvedores de aplicativo educacional, bem como ajudam o aprendiz a manter o foco no conteúdo estudado.

Tabela 5 – Diretrizes para produção de mídia.

DESIGN DE CONTEÚDO M-LEARNING				
#	Recomendação	Descrição	Referência	Responsabilidade (D - Didático / T - Técnico)
TEXTO				
I	Facilite a compreensão	A ideia nessa diretriz é produzir um conteúdo educacional que mantenha o foco no que realmente é relevante para aprendizagem, evitando a presença no texto de informações desnecessárias ao entendimento de um conceito.	(CHITTARO, 2010) (NIELSEN; BUDIÚ, 2013) (KLEPSCH; SEUFERT, 2020)	D/T
II	Tamanho da linha de texto	Segundo diretrizes propostas por fabricantes de sistemas operacionais para dispositivos móveis, 60 caracteres por linha é uma quantidade confortável para leitura e garante uma boa legibilidade. Textos longos que demandam mais tempo de leitura podem causar desconforto aos usuários.	(CHITTARO, 2010) (GOOGLE, 2018) (APPLE, 2018)	D/T
III	Escolha de fonte e cores	A variação de tamanho de fonte e cores é uma forma de enfatizar itens importantes no aplicativo, portanto os autores devem lançar mão desse recurso para enfatizar trecho relevantes do texto.	(APPLE, 2018)	D/T
IV	Controle de quebra de linha	Na quebra de linha evite deixar grandes lacunas, preposições ou palavras curtas no seu fim. Uma organização textual adequada evita a necessidade de uso de de palavras, melhorando assim a fluidez da leitura.	(GOOGLE, 2018)	T
V	Utilize sensores dos dispositivos móveis	Os recursos sensoriais dos dispositivos móveis (captura de áudio, vídeo, GPS, câmera) podem ser explorados para auxiliar a entrada de dados, como em campos textos que podem ser preenchidos por reconhecimento de voz ou capturados de dados por meio da câmera.	(GOOGLE, 2018) (APPLE, 2018)	T
VI	Legibilidade de texto sobre imagem	Para que a legibilidade de texto sobre imagens não seja prejudicada, aplique uma camada de gradiente. O gradiente deve ter uma opacidade que torne a legibilidade do texto natural e não prejudique a visibilidade da imagem.	(GOOGLE, 2018)	D/T
IMAGEM				
I	Atenção a estética e relevância	A imagem deve retratar o contexto com beleza estética e relevância. Utilize imagens que tornem a compreensão fácil e imediata, esse cuidado diferencia o conteúdo educacional e melhora a experiência do aprendiz.	(APPLE, 2018) (MAYER; FIORELLA; STULL, 2020)	D
II	Faça uso de ilustrações e fotografia	As fotografias são melhores aproveitadas para expressar entidades e história específicas, enquanto que a ilustração representar melhor conceitos e metáforas.	(APPLE, 2018)	D
III	Mantenha um foco icônico na imagem	Inclua na imagem elementos que permitam que o aprendiz tenha uma compreensão clara da mensagem com pouca carga cognitiva.	(APPLE, 2018)	D/T
IV	Crie imagens com uma construção narrativa	Imagens que contam uma história são mais informativa e compreensíveis.	(GOOGLE, 2018)	D
V	Atenção a dimensões e resolução da imagem	Produza imagens com dimensões adequadas ao tamanho de tela e plataformas.	(GOOGLE, 2018) (APPLE, 2018)	D/T
VI	Utilize rótulos de texto	Atribua rótulo de texto a imagens para que os aprendizes possam se beneficiar do recurso de leitura de tela.	(APPLE, 2018)	D/T
VÍDEO				
I	Duração do vídeo	Conforme pesquisas e plataformas de visualização de vídeos, os usuários sentem-se mais confortáveis visualizando vídeos com duração entre 5 a 10 minutos.	(DIWANJI <i>et al.</i> , 2014) (GOOGLE, 2018)	D
II	Segmente o conteúdo	A divisão do conteúdo em porções menores melhora a retenção do conhecimento.	(DIWANJI <i>et al.</i> , 2014) (BIARD; COJEAN; JAMET, 2017) (MAYER, 2017) (MYLLYMÄKI; HAKALA; HÄRMÄNMAA, 2018)	D
III	Conteúdo relevante ao curso	Os vídeo que apresentam conteúdo diretamente relacionados ao conteúdo programático do curso são mais visualizados.	(DIWANJI <i>et al.</i> , 2014) (MAYER; FIORELLA; STULL, 2020)	D
IV	Qualidade de edição do vídeo	A preocupação com uma aparência profissional com recursos e visual de qualidade torna o vídeo mais envolvente.	(DIWANJI <i>et al.</i> , 2014) (MADARIAGA <i>et al.</i> , 2021)	D/T
V	Mantenha um discurso envolvente	Produza vídeo com um estilo de diálogo mais leve e bem humorado e mantenha uma velocidade do discurso adequada de forma a não causar monotonia no aprendiz.	(DIWANJI <i>et al.</i> , 2014)	D/T

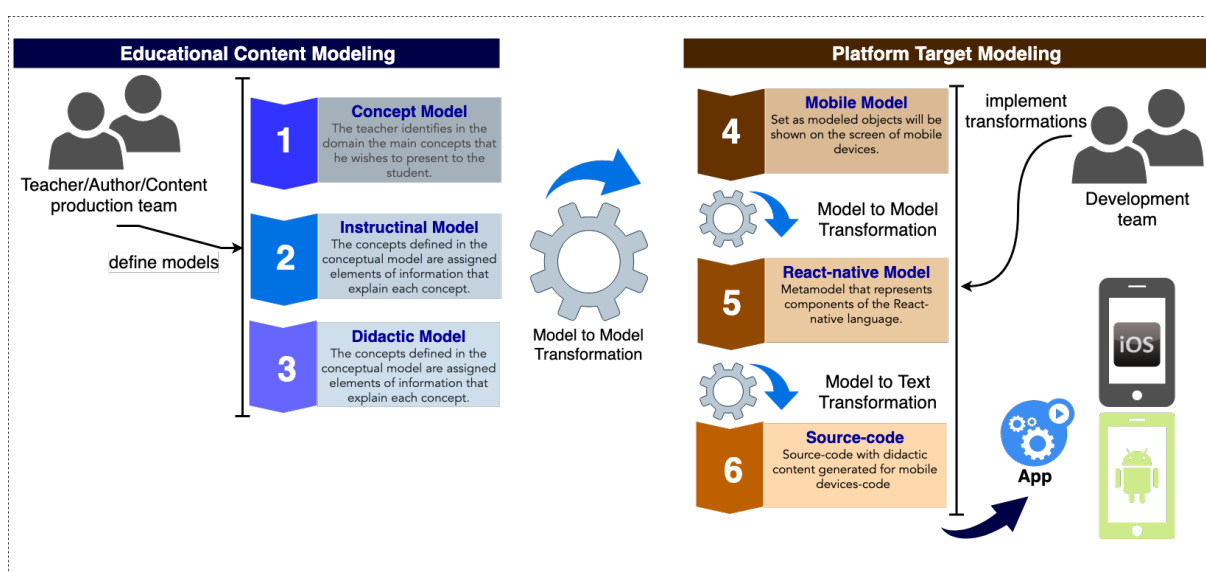
3.3 Modelagem do conteúdo educacional

A Figura 20 apresenta uma visão geral dos modelos construídos nos dois espaços tecnológicos estabelecidos pela EDUCOM.ML. No espaço tecnológico que representa a *Educational Content Modeling*, a equipe de produção de conteúdo constrói instâncias dos modelos Conceitual, Instrucional e Didático, estabelecendo como os conceitos e elementos instrucionais do domínio do conhecimento são estruturados. Já no espaço tecnológico *Platform Target Modeling*, a equipe de desenvolvimento constrói instâncias de modelos que representam aspectos técnicos da plataforma alvo de implementação do conteúdo educacional. Destaca-se, que a equipe de desenvolvimento implementa regras de transformação de modelos para instanciar os modelos da *Platform Target Modeling* a partir dos modelos definidos na *Educational Content Modeling* pelos produtores de conteúdo educacional.

A proposta de organizar a produção do conteúdo educacional em duas visões proporciona uma liberdade na escolha do ambiente de estudo que se pretende apresentar o conteúdo educacional. Isto porque, a visão da *Educational Content Modeling* não exige da equipe de produção o domínio de aspecto de implementação da tecnologia envolvida no desenvolvimento. Assim, a equipe de produção mantém o foco em estruturar o conteúdo educacional, enquanto a equipe de desenvolvimento tem a responsabilidade de implementar os metamodelos e regras de transformação que apresentam o conteúdo modelado em diferentes ambientes de estudo (web, móvel, PDF, apresentação). Logo, um único conteúdo educacional modelado pode ser utilizado para a produção de conteúdo didático para diversas plataformas de ensino e aprendizagem.

Nas seções seguintes os metamodelos que atendem os dois espaços tecnológicos previstos pela EDUCOM.ML são detalhados.

Figura 20 – Visão geral da EDUCOM.ML.



Fonte: elaborada pelo autor.

3.3.1 Metamodelo Educom

O Metamodelo Educom representa os conceitos estabelecidos pela AIM-CID e visa criar instâncias dos modelos Conceitual, Instrucional e Didático, estruturando assim o conteúdo educacional do domínio do conhecimento sem considerar aspectos da tecnologia de sua implementação. A Figura 21 ilustra os elementos do MetaModelo Educom (MME). Os elementos estão em cores distintas para identificar cada modelo representado no MME. o Modelo Conceitual é composto por elementos na cor vermelha, elementos na cor verde e cinza pertencem ao Modelo Instrucional e elementos na cor azul compõem o Modelo Didático. Existe, ainda, elementos na cor laranja que representam o Modelo Mídias. O Modelo de Mídia fornece classes que visam representar textos, imagens, vídeo, áudio e link.

Modelo Conceitual

Seguindo as etapas apresentadas na Figura 20, a primeira tarefa da equipe de produção é identificar quais conceitos do domínio do conhecimento serão modelados, isto é, construir uma instância do Modelo Conceitual. As as classes IKNOWLEDGEDOMAIN, MODULE, ICONCEPT e IREFERENCE são instanciadas para a construção do Modelo Conceitual. A seguir essas classes são detalhadas:

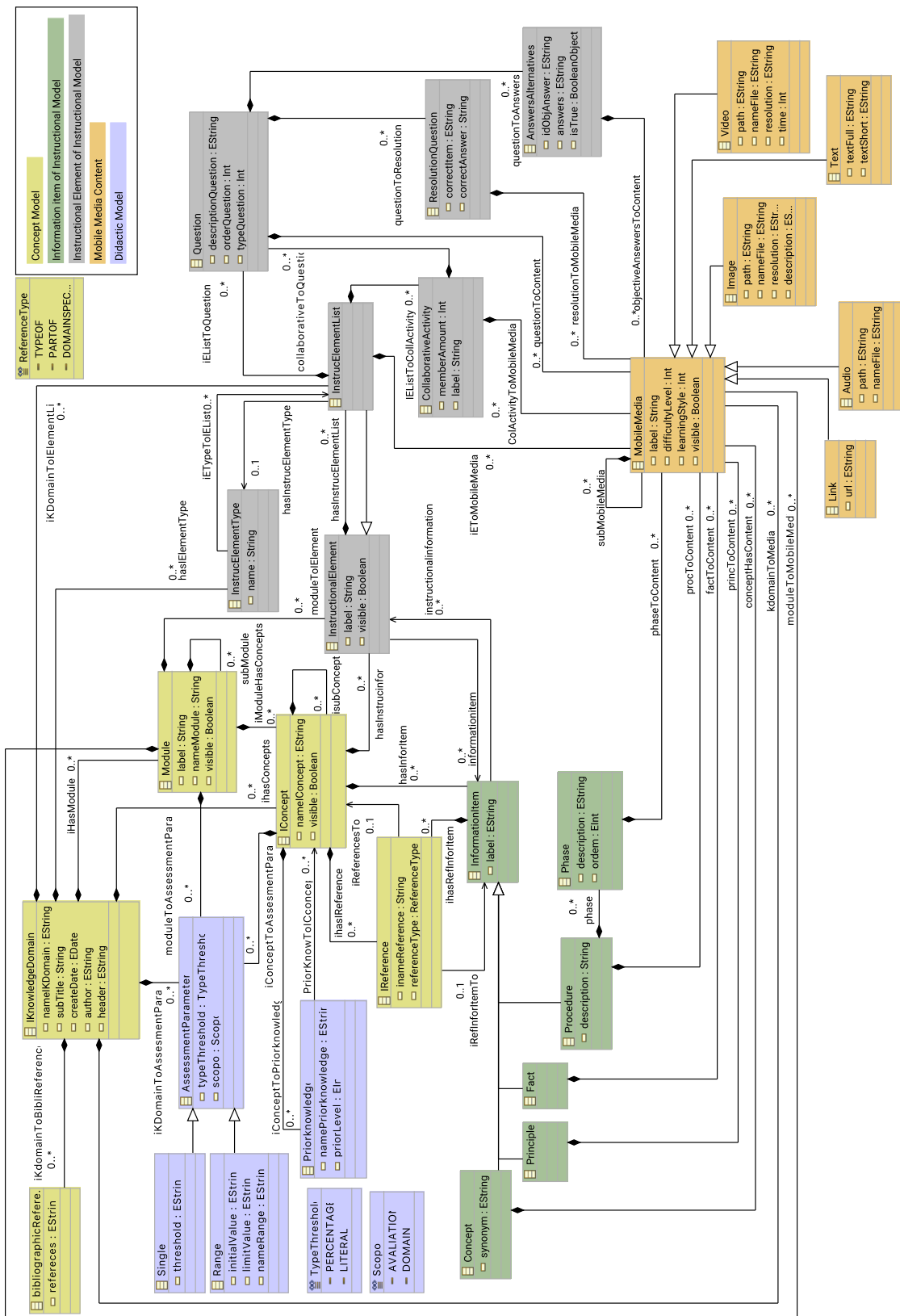
- IKNOWLEDGEDOMAIN: tem a responsabilidade de registrar dados sobre o domínio do conhecimento que está sendo modelado. Informações como nome do domínio, autor, data de sua criação podem ser informadas pela equipe de produção. O MME permite a segmentação do domínio pela criação de instâncias de um conjunto de módulos (MODULE), porém, é possível também que sejam modelados conceitos (ICONCEPT) desassociados aos módulos do domínio. O MME estabelece ainda um relacionamento da classe IKNOWLEDGEDOMAIN com a classe BIBLIOGRAPHREFERENCES, a finalidade dessa associação é registrar referências bibliográficas utilizadas na produção do conteúdo educacional. A Tabela 6 detalha os atributos da classe IKNOWLEDGEDOMAIN.

Tabela 6 – Classe IKnowledgeDomain

Atributo	Descrição
nameDomain	Nome do domínio modelado.
subTitle	Nome auxiliar ou breve descrição do domínio modelado. Atributo opcional.
createDate	Data de criação do modelo atual. Mantém um controle de versão do modelo.
author	Autores reensáveis pela construção do conteúdo educacional.
header	Atributo destinado à identificação de cabeçalhos de documentos gerados. Por exemplo, se o conteúdo criado faz parte de curso ou instituição de ensino.

- MODULE: tem a finalidade de representar os módulos de ensino do domínio modelado. Como sugerido na construção do Modelo Conceitual, o conteúdo educacional deve ser segmentado em módulos concisos de ensino. Módulos ainda podem ser compostos por outros módulos, dando ao time de produção a possibilidade de segmentação refinada do

Figura 21 – Metamodelo Educom.



Fonte: elaborada pelo autor.

conteúdo educacional. Além de um auto-relacionamento, a classe MODULE se associa por composição com as classes ICONCEPT, INSTRUCTIONALELEMENT, MOBILEMIDIA e ASSESSMENTPARAMETERS. A Tabela 7 detalha os atributos da classe MODULE.

Tabela 7 – Classe Module

Atributo	Descrição
label	Tem a finalidade de registrar um identificador do módulo. Esse atributo é utilizado para processamento interno de regras de transformação do modelo EDUCOM.ML em outros modelos.
nameModule	Atributo que registra o nome do módulo. Esse atributo é utilizado para apresentar o nome do módulo no conteúdo educacional gerado.
visible	Atributo tem a finalidade de configurar a visibilidade do módulo. Por padrão todo módulo é visível e estará presente no conteúdo educacional gerado. No entanto, ao criar o Modelo Didático a equipe de produção do conteúdo pode definir quais módulos serão apresentados ao aprendiz setando o valor do atributo para verdadeiro ou falso.

- **ICONCEPT**: representa os conceitos definidos para cada módulo do domínio do conhecimento. No Modelo Conceitual, apenas os conceitos relevantes para o domínio que se pretende ensinar são modelados. Assim, informações mais detalhadas de cada conceito (ICONCEPT) não são modeladas. Por representar somente os conceitos, a classe ICONCEPT é composta apenas por dois atributos: i) o atributo **attr::nameConcept** que registra o nome do conceito. ii) e o atributo **attr::visible** que define se o conceito será apresentado ao aprendiz. O atributo **attr::visible** é definido por padrão como verdadeiro, mas seu valor pode ser ajustado conforme a estratégia utilizada na construção do Modelo Didático. Como propõe o Modelo Conceitual, os conceitos do domínio são instanciados considerando uma hierarquia de relacionamento existentes entre eles. No MME essas relações são representadas pela classe IREFERENCE, composta por um atributo que nomeia a relação (**attr::inameReference**) e outro que classifica a relação (**attr::referenceType**). As associações entre conceitos podem ser classificadas como *typeOf*, *partOf* e *domainSpecific*.

Em linhas gerais, a modelagem do domínio do conhecimento tem início com a criação de instâncias das classes IKNOWLEDGEDOMAIN, MODULE, ICONCEPT, IREFERENCE (Modelo Conceitual). Essas classes possibilitam a equipe de produção de conteúdo identificar e construir uma estrutura hierárquica de conceitos que guiará toda a produção do conteúdo educacional de um domínio modelado.

Modelo Instrucional

Com os conceitos relevantes do domínio definidos no Modelo Conceitual, a tarefa seguinte é incorporar aos conceitos elementos que os expliquem e reforcem o aprendizado. Para atender essa tarefa, o MME fornece classes para implementar o Modelo Instrucional. Elementos que tornam compreensíveis cada conceito são representados por instâncias da classe INFORMATIONITEM e elementos que reforçam o aprendizado são representados pela classe INSTRUCTIONALELEMENT. A seguir essas classes são detalhadas.

- **INFORMATIONITEM**: tem a finalidade de categorizar as informações adicionais que descrevem os conceitos conforme a teoria *Component Display Theory*–CDT (MERRILL, 1983). Deste modo, para representar Itens de Informação foram criadas as classes **CONCEPT**, **PRINCIPLE**, **PROCEDURE** e **FACT**. Basicamente, essas classes são compostas por um atributo (**attr::label**) que identifica o conceito referenciado no Item de Informação modelado. A exceção são as classes **PROCEDURE**, que possui um atributo adicional para manter uma descrição (**attr::description**) do procedimento modelado, e a classe **PHASES**, que registra as fases envolvidas em um procedimento, composta pelos atributos **attr::description** e **attr::ordem** (registra a ordem de execução das fases). Um **INFORMATIONITEM** ainda pode associar-se com um **INSTRUCTIONALELEMENT**, criando assim uma estratégia de definição de elemento de informação para reforçar a aprendizagem de um conceito.
- **INSTRUCTIONALELEMENT**: permite que recursos como exercícios, exemplos, uso de ferramentas, avaliações e atividades colaborativas sejam atribuídas aos conceitos ou módulos do domínio. Um **INSTRUCTIONALELEMENT** dispõe de um atributo que o nomeia (**attr::label**) e outro que determina sua visibilidade (**attr::visible**). Da mesma forma de outros elementos do MME, o **attr::visible** por padrão é verdadeiro, porém ao construir o Modelo Didático sua configuração pode ser ajustada ao objetivo da aprendizagem. A atribuição de valor verdadeiro ou falso para o atributo **attr::visible** permite que o autor determine quais objetos modelados farão parte do conteúdo educacional gerado. As classes a seguir dão apoio à definição de Elementos Instrucionais:
 - **INSTRUCELEMENTTYPE**, tem a finalidade de classificar um Elemento Instrucional. A classificação é determinada pela equipe de produção de conteúdo, os mais comuns são exercícios, exemplos, avaliações, atividades colaborativas. Essa classificação guia a implementação de regras de transformação do Modelo Educom em modelos da plataforma alvo de exibição do conteúdo educacional, permitindo que interfaces adequadas a cada categoria de elemento instrucional sejam implementadas.
 - **INSTRUCELEMENTLIST**, mantém uma lista de Elementos Instrucionais definidos para um dado conceito. Os seus atributos são herdados da classe **INSTRUCTIONALELEMENT**.
 - **COLLABORATIVEACTIVITY**, mantém as atividades colaborativas elaboradas pela equipe de produção de conteúdo. A classe **COLLABORATIVEACTIVITY** é composta pelo atributo **attr::label**, que nomeia a atividade, e pelo atributo **attr::memberAmount**, que determina a quantidade de membros que participam da atividade. Além disso, a classe **COLLABORATIVEACTIVITY** tem relacionamento com as classes **MOBILEMEDIA** e **QUESTION**, proporcionando o uso de recursos de mídia em uma atividade colaborativa, questionários e avaliações.
 - **QUESTION**, mantém exercícios e avaliações que envolvem questões subjetivas e objetivas. Para cada questão definida para módulos e conceitos são mantidas

as informações de enunciado da questão (**attr::descriptionQuestion**), sua ordem (**attr::orderQuestion**) e o tipo de questão (**attr::typeQuestion**). O atributo (**attr::typeQuestion**) classifica as questões em subjetivas ou objetivas. Para questões objetivas, um relacionamento com a ANSWERSALTERNATIVES é mantido para definir as alternativas que serão apresentadas ao aprendiz. Ademais, existe um relacionamento com a classe RESOLUTIONQUESTION para registrar a resolução de uma questão.

- ANSWERSALTERNATIVES, mantém as alternativas de questões classificadas como objetivas. A classe registra um identificador da alternativa (**attr::idObjAnswer**), sua descrição (**attr::answers**) e se a alternativa é a resposta correta (**attr::isTrue**) da questão.
- RESOLUTIONQUESTION, mantém a resolução de questões subjetivas elaboradas para um módulo, conceito ou atividade colaborativa. O objetivo de manter a resolução de questões subjetivas é preparar o MME para permitir a implementação de algoritmos que ajudem na correção automática.

Além de permitir a definição de Itens de Informação e Elementos Instrucionais, o MME fornece a classe MOBILEMEDIA que cria instâncias de textos, imagens, vídeos, áudios e links utilizados na produção do conteúdo. A classe MOBILEMEDIA é utilizada pelos modelos Conceitual e Instrucional para especificar elementos de mídia. A classe fornece, ainda, atributos que identificam nível de dificuldade (**attr::difficultyNivel**) e estilo de aprendizagem (**attr::learningStyle**) das mídias produzidas estão disponíveis. Esses atributos fornecem à equipe de produção a possibilidade de identificar para qual estilo de aprendizagem uma mídia foi produzida. Essa informação setada, cria as condições necessárias para implementação de recursos de gerenciamento da aprendizagem. Detalhes da classe MOBILEMEDIA são apresentadas a seguir:

- TEXT, classe utilizada para atribuição de textos aos Itens de Informação e Elementos Instrucionais. O atributo **attr::textFull** pode ser utilizado pela equipe de produção para uma descrição mais detalhada do conceito ou elemento do MME. Enquanto o atributo **attr::textShort** é utilizado para uma descrição mais resumida de um conceito. É desejável que a equipe de produção utilize os dois atributos, assim, o conteúdo educacional produzido pode se adequar a objetivos diferentes de ensino. Por exemplo, caso a intenção da geração do conteúdo educacional seja apresentar um material didático mais aprofundado, o conteúdo do atributo **attr::textFull** é utilizado, caso contrário, a intenção seja apresentar um conceito sucintamente, o atributo **attr::textShort** será utilizado.
- IMAGE, mantém registros relacionados às características de imagens utilizadas no conteúdo modelado. Informações como nome do arquivo e caminho de armazenamento são alguns dos atributos armazenados. A Tabela 8 destalha todos os atributos desta classe.

Tabela 8 – Classe Image

Atributo	Descrição
path	caminho de localização de armazenamento do arquivo de imagem.
nameFile	nome do arquivo de imagem.
resolution	resolução da imagem armazenada.
description	descrição utilizada para ser apresentada juntamente com a imagem no conteúdo educacional.

- VIDEO, mantém registro de características de vídeos atribuídos ao conteúdo educacional modelado. Destalhes dos atributos que compõe esta classe são apresentados na Tabela 9.

Tabela 9 – Classe Video

Atributo	Descrição
path	caminho da localização do arquivo de vídeo
nameFile	nome do arquivo de vídeo.
resolution	qualidade de exibição do vídeo (480p, 720p, 1080p...)
proportion	formato de tela do vídeo (4:3, 16:9...).
time	tempo de duração do vídeo.

- AUDIO, classe que mantém informações sobre arquivo de áudio utilizado na produção do conteúdo educacional. A classe é composta por um atributo que registra o caminho de armazenamento (**attr::path**) do arquivo de áudio, um atributo com o nome do arquivo (**attr::nameFile**) e um atributo que armazena o tempo (**attr::time**) de áudio do arquivo.

Em síntese, é no Modelo Instrucional que elementos que ajudam o aprendiz a compreender cada conceito são modelados. Vale ressaltar que o planejamento e projeto do MME não impõe uma estratégia pedagógica específica. O MME foi projetado para dar à equipe de produção a liberdade de definir a estratégia pedagógica que se alinhe melhor ao objetivo do ensino e aprendizagem.

Modelo Didático

O Modelo Didático tem a função de estabelecer a navegabilidade entre as informações definidas no Modelo Instrucional. Com a hierarquia de conceitos definidos no Modelo Conceitual e com as informações complementares atribuídas no Modelo Instrucional, a tarefa seguinte é estabelecer a navegabilidade e visibilidade do conteúdo educacional. Para tanto, a equipe de produção precisa estabelecer os conhecimentos prévios necessários para estudo de um determinado conceito e, ainda, escolher quais módulos, conceitos e elementos instrucionais serão visíveis ao aprendiz. A definição de visibilidade e conhecimentos prévios não é uma tarefa obrigatória e depende dos objetivos educacionais pretendidos com o conteúdo modelado.

Para atender os requisitos de navegabilidade do conteúdo educacional projetado, o MME disponibiliza as classes ASSESSMENTPARAMETERS e PRIORKNOWLEDGE. A classe ASSESSMENTPARAMETERS permite definir parâmetros que serão utilizados para determinar se

o aprendiz atingiu o aprendizado desejado em um conceito, módulo ou atividade avaliativa. O parâmetro de avaliação pode ter como referência um valor unitário ou uma faixa de valores. Valor unitário é definido com instância da classe SINGLE, enquanto faixas de valores são definidas com instâncias da classe RANGE. A Tabela 10 detalha os atributos das classes envolvidas na definição de parâmetro de avaliação.

Tabela 10 – Classes que suportam a definição de parâmetro de avaliação.

Atributo	Descrição
Class::AssessmentParameters	
typeThreshold	define o valor de referência da avaliação como uma porcentagem ou valor absoluto. Essa classificação é referenciada por uma enumeração (enum::TypeThreshold) com os valores PERCENTAGEM e LITERAL.
scope	atributo que determina o escopo em que os parâmetros de avaliação serão utilizados. Quando o escopo é setado como AVALIATION, a referência de avaliação considera as atividades avaliativas definidas no conteúdo educacional produzido. E quando o escopo é setado como DOMAIN, a referência é o percentual de conteúdo programático já estudado pelo aprendiz.
Class::Single -> extend::AssessmentParameters	
threshold	atributo que registra o valor de referência que determina que o aprendiz atingiu um índice de aprendizagem desejável.
Class::Range -> extend::AssessmentParameters	
initialValue	valor inicial da faixa de valores que determina o índice de aprendizagem desejável do aprendiz.
limitValue	valor final da faixa de valores que determina o índice de aprendizagem desejável do aprendiz.

A classe PRIORKNOWLEDGE representa o conhecimento prévio que precisa ser aprendido para que um conceito seja estudado. Assim, essa classe oferta recursos para orientar o caminho de estudo do aprendiz. O atributo **attr::priorLevel** determina se um conhecimento prévio é obrigatório ou desejável, possibilitando assim a flexibilização do caminho escolhido para estudo do domínio. Caso no projeto de conteúdo educacional não sejam determinados conhecimentos prévios para os conceitos, a navegabilidade entre eles é estabelecida pelo aprendiz durante o uso do conteúdo educacional.

O Metamodelo Educom foi projetado para atender as necessidades de estruturação do conteúdo educacional. No escopo desse trabalho, a preocupação é produzir conteúdo para atender as necessidades de aplicativos educacionais móveis. Portanto, algumas diretrizes foram propostas com esse objetivo. No entanto, o MME pode ser utilizado para projetar conteúdo educacional para outros escopos, como plataformas web, criação de apresentação e construção de documentos PDF. Contudo, essa mudança de escopo deve ser acrescida de diretrizes que condução a produção do conteúdo atendendo às características de cada plataforma alvo. Na seção seguinte, os metamodelos propostos para implementação da plataforma alvo para apresentação do conteúdo educacional são detalhados.

3.4 Modelagem da Plataforma Alvo

3.4.1 Metamodelo Mobile

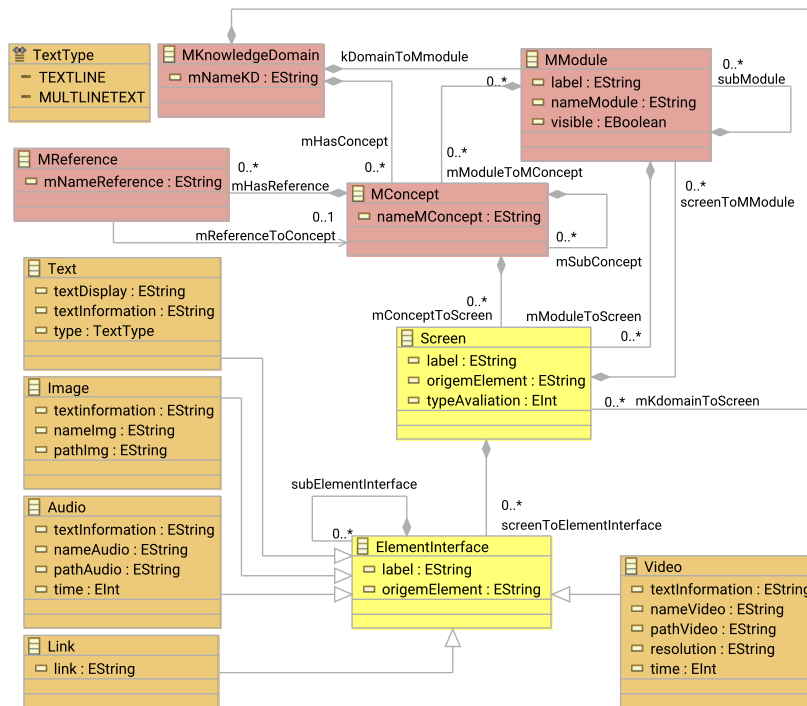
O MME estrutura o conteúdo educacional do domínio, no entanto, não é de sua responsabilidade projetar a estrutura de telas da aplicação nas quais serão apresentados os módulos, conceitos, itens de informação e elementos instrucionais. Essa tarefa de determinar a distribuição do conteúdo educacional modelado entre as telas da aplicação móvel é desempenhada, na abordagem EDUCOM.ML, pelo Metamodelo Mobile. O intuito dessa estratégia é criar uma representação abstrata da distribuição do conteúdo educacional desconsiderando aspectos da tecnologia utilizada na implementação da aplicação móvel.

Um Modelo Mobile é obtido de forma automática pela aplicação de regras de transformação de modelos. Portanto, regras de transformação são aplicadas ao Modelo Educom para geração do Modelo Mobile. Em linhas gerais, a regras de transformação do Modelo Educom em Modelo Mobile aplicam a seguinte lógica: i) os módulos definidos no Modelo Educom serão apresentados em uma única tela; ii) os conceitos de um determinado módulo serão exibidos em uma tela separada; iii) que cada conceito terá uma tela para exibição de seu conteúdo educacional.

O Metamodelo Mobile tem como base a hierarquia de conceitos definidos no Modelo Conceitual do MME. Além disso, foram criadas classes para representar elementos de mídias definidos no Modelo Instrucional do MME. As classes que constituem o Metamodelo Mobile são ilustradas na Figura 22 e o detalhamento dos seus atributos são apresentados seguir.

- **MKNOWLEDGEDOMAIN**, tem a finalidade de representar o domínio do conhecimento modelado, composta pelo atributo **attr::mNameKD** que registra o nome do domínio.
- **MMODULE**, representa os módulos definidos para um domínio. Uma instância da classe **MMODULE** registra o nome do módulo (**attr::nameModule**) e sua visibilidade (**attr::visible**). O atributo **attr::visible** setado para verdadeiro indica que o módulo estará presente no aplicativo educacional móvel gerado.
- **MCONCEPT**, representa os conceitos definido para cada módulo. A classe é constituída do atributo **attr::nameMConcept**.
- **SCREEN**, essa classe representa uma abstração das telas da aplicação móvel. A classe é composta pelos seguintes atributos:
 - **attr::label**, cria um rótulo de identificação da tela instanciada.
 - **attr::origemElement**, identifica o elemento do Modelo Educom que está sendo representado pela classe **SCREEN**.
 - **attr::typeAvaliation**, atributo que indica se a tela instanciada apresenta uma avaliação ou exercício. O atributo ainda determina se a avaliação é formada por uma questão

Figura 22 – Metamodelo Mobile.



Fonte: elaborada pelo autor.

objetiva ou subjetiva. Essa informação é útil na construção de interface de usuário mais adequada a cada tipo de questão.

- **ELEMENTINTERFACE**, classe que generaliza mídias como texto, imagem, áudio, link e vídeo utilizados na modelagem do conteúdo educacional. **ELEMENTINTERFACE** tem o atributo rótulo (**attr::label**), que nomeia o elemento de interface e o atributo **attr::origemElement** que faz referência a um elemento do Modelo Educom.
- **TEXT**, representa mídia texto utilizado na modelagem do conteúdo educacional. A classe fornece o atributo **attr::textDisplay**, que registra o texto que será exibido na tela da aplicação, e o atributo **attr::type**, que classifica o texto como “de única linha” ou “de múltiplas linha“. Essa atributo orienta a implementação o comportamento do elemento de interface na aplicação móvel.
- **IMAGE**, classe que registra imagens utilizadas na modelagem do conteúdo educacional. A Tabela 11 detalha os atributos da classe.
- **VIDEO**, registra vídeo atribuído à modelagem de conceitos. A Tabela 12 detalha os atributos da classe.
- **AUDIO**, registra áudios definidos para apresentar conceitos ou reforçar a aprendizagem. A Tabela 13 detalha seus atributos.

Tabela 11 – Classe Image

Atributo	Descrição
textInfomation	Texto com a descrição da imagem que pode ser utilizado na implementação da interface da tela da aplicação móvel.
pathImage	caminho da localização do arquivo de imagem.
nameImage	nome do arquivo de imagem.
resolution	qualidade de exibição da imagem.

Tabela 12 – Classe Video

Atributo	Descrição
textInfomation	Texto com a descrição do vídeo que pode ser utilizado na implementação da interface da tela da aplicação móvel.
pathVideo	caminho da localização do arquivo de vídeo
nameVideo	nome do arquivo de vídeo.
resolution	qualidade de exibição do vídeo (480p, 720p, 1080p...)
proportion	formato de tela do vídeo (4:3, 16:9...).
time	tempo de duração do vídeo.

Tabela 13 – Classe Audio

Atributo	Descrição
textInfomation	Texto com a descrição do áudio que pode ser utilizado na implementação da interface da tela da aplicação móvel.
pathAudio	caminho da localização do arquivo de vídeo
nameAudio	nome do arquivo de áudio.
time	tempo de duração do áudio.

- LINK, classe utilizada para registrar link para aprofundamento do estudo de um conceito. A classe possui o atributo **attr:link**, que armazena o endereço de acesso ao conteúdo de aprofundamento.

Em suma, o Modelo Mobile apresenta uma visão abstrata, isto é, independente de tecnologia de implementação, dos componentes que estarão presentes nas interfaces da aplicação móvel gerada com a abordagem EDUCOM.ML. O próximo passo no processo de geração do conteúdo educacional é transformar o Modelo Mobile em um modelo específico da plataforma móvel alvo. Nesse trabalho a aplicação educacional móvel será implementada com o *framework* React-native. A seção seguinte detalha os metamodelos para a linguagem programação React-native.

3.4.2 Metamodelo React-native (MRn)

A necessidade de implementar código-fonte específico para cada plataforma móvel disponível no mercado é uma dificuldade presente na implementação de aplicações móveis (EISENMAN, 2017). Uma estratégia que minimiza essa dificuldade é a utilização de abordagens com o recurso de criar um único código-fonte e gerar aplicativos de plataforma cruzada (Latif *et al.*, 2017). Uma dessas abordagens é o *framework* React-native (RN) (FACEBOOK, 2020),

que implementa aplicativos móveis que podem ser utilizados nas plataformas Android e IOS. Essa capacidade do RN de implementar aplicativos multiplataforma foi o requisito considerado na sua adoção como linguagem de programação (LP) para implementação da aplicação móvel gerada pela abordagem de modelagem EDUCOM.ML.

Com o objetivo de gerar aplicativos móveis para apresentação do conteúdo educacional modelado, um Metamodelo React-native (MRN) foi proposto. O MRN foi modelado considerando dois conjuntos de componentes, o primeiro conjunto é responsável por implementar classes que representam componentes que permitem a geração de código-fonte relacionado ao *framework* RN. O segundo conjunto de componentes é responsável por implementar classes que criam uma estrutura de dados para armazenar o conteúdo educacional modelado pela EDUCOM.ML. O particionamento de responsabilidades na modelagem do MRN visa fornecer flexibilidade ao processo de geração do conteúdo educacional. Ao dividir o metamodelo em elementos que representam conteúdo e elementos que representam componentes da LP, o MRN cria condições de reutilização do conteúdo educacional em aplicações móveis implementadas com outra linguagem de programação.

O Modelo RN é construído de forma automática pela aplicação de regras de transformação. As regras de transformação do Modelo Mobile em Modelo RN foram construídas obedecendo a modelagem de telas definidas no Modelo Mobile. Assim, toda a hierarquia de elementos do conteúdo educacional projetado no Modelo Mobile resulta na criação de instância do Modelo RN. Como ilustra a Figura 23, o MRN tem seus elementos representados no metamodelo em cores distintas. Os elementos na cor verde representam componentes do *framework* RN, enquanto elementos na cor cinza representam a estrutura de classes utilizadas na implementação do conteúdo educacional.

Neste documento, classes do MRN que representam o conjunto de elementos que implementam componentes do *framework* RN não são detalhados. Isto porque informações sobre cada um deles podem ser encontradas em sua documentação oficial ¹. Deste modo, apenas os elementos que estão diretamente ligados à implementação do conteúdo educacional modelado com a EDUCOM.ML são detalhados. A seguir as classes que dão apoio à estruturação do conteúdo educacional são apresentadas:

- PROJETORN, é a primeira classe instanciada na construção de um Modelo RN. A classe tem o objetivo de identificar o domínio que está sendo modelado, composto pelo atributo **attr::name** para registro do projeto.
- DATAJS, classe responsável em manter os dados relacionados ao conteúdo educacional projetado. Um DATAJS tem um relacionamento de composição com classes que representam mídias, módulos e conceitos do domínio modelado.

¹ <https://reactnative.dev>

- ARRAY, classe que faz referência ao conjunto de *array* criados para armazenar o conteúdo educacional. Um ARRAY pode ser composto por instâncias das classes OBJECTELEMENT, ELEMENTIMAGE ou ELEMENTTEXT.
- OBJECTELEMENT, representa abstração dos elementos do conteúdo educacional (módulos, conceitos, itens de informação e elementos instrucionais), além de indicar se o conteúdo será exibido na aplicação móvel. A visibilidade de um objeto é definida pelo atributo **attr::visible**.
- ELEMENTIMAGE, representa elementos imagem que são modelados no conteúdo educacional.
- ELEMENTTEXT, representa elementos texto que são modelados no conteúdo educacional.
- PRIORKNOWLEDGE, classe que define o conhecimento prévio necessário para o estudo de um dado conceito.
- PARAMETERAVALIATION, classe responsável em indicar a medida que irá determinar se o aprendiz atingiu o nível de aprendizagem desejada de um conceito. A medida pode ser um valor único ou uma faixa de valores. O atributo **attr::avaliationType** categoria o parâmetro de avaliação em simples ou faixa de valor.

Resumidamente, ao instanciar o MRN criam-se os elementos necessários para a geração do código-fonte RN da aplicação móvel que apresentará o conteúdo educacional modelado. Esse código-fonte é obtido pela transformação de Modelo para Texto, isto é, regras de transformação são aplicadas ao Modelo RN obtendo-se o código-fonte que a equipe de desenvolvimento utilizará para implementar o aplicativo educacional móvel. Ressalta-se que o MRN proposto não representa todos os componentes da *framework* RN; foram considerados para esse trabalho apenas os relevantes para a geração da aplicação móvel modelada pela abordagem EDUCOM.ML.

3.5 Ferramenta de autoria Educom Tool

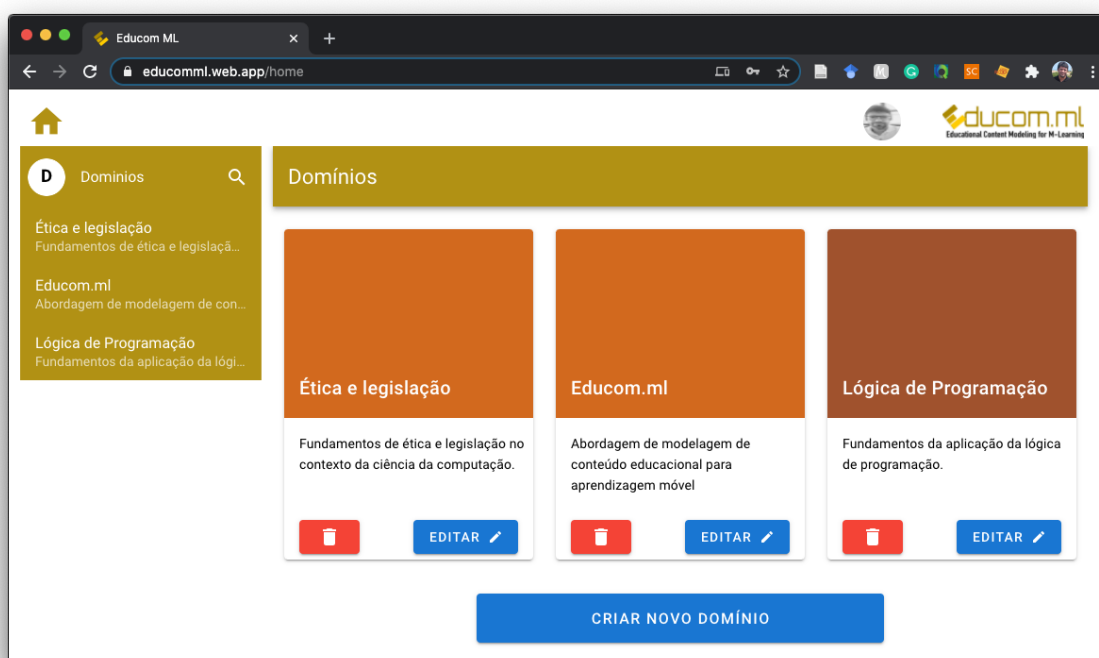
Na implementação da abordagem EDUCOM.ML os modelos se concretizam por instâncias construídas com o *Eclipse Modeling Framework* – EMF. Ocorre que o ambiente do EMF está próximo das habilidades dos profissionais de desenvolvimento e distante dos conhecimentos dos professores em trabalhar com ferramenta de modelagem. Nesse contexto, o conteúdo educacional seria elaborado pelos professores e repassado aos desenvolvedores para a construção dos modelos Conceitual, Instrucional e Didático. Visando permitir que os modelos Conceitual, Instrucional e Didático sejam construídos pelos professores um protótipo da ferramenta EDUCOM TOOL ² foi desenvolvido. A finalidade da EDUCOM TOOL é fornecer aos professores um ambiente adequado

² Página web com acesso a Educom Tool: educomml.web.app

para elaboração do conteúdo educacional utilizado para instanciar os modelos Conceitual, Instrucional e Didático.

A Figura 24 apresenta a interface de entrada da EDUCOM TOOL. Nesta interface o professor pode escolher criar um novo domínio ou acessar os já modelados. Caso a opção “criar novo domínio” seja selecionada, o ambiente de definição de módulos, conceitos, elementos instrucionais e itens de informação é apresentado ao professor. Caso a escolha seja pela opção “editar” os módulos, conceitos, elementos instrucionais e itens de informação que já foram modelados são apresentados ao professor. Vale destacar que apenas os domínios de um dado usuário são apresentados nessa interface.

Figura 24 – Interface para iniciar a modelagem de domínio do conhecimento da aplicação EDUCOM TOOL.



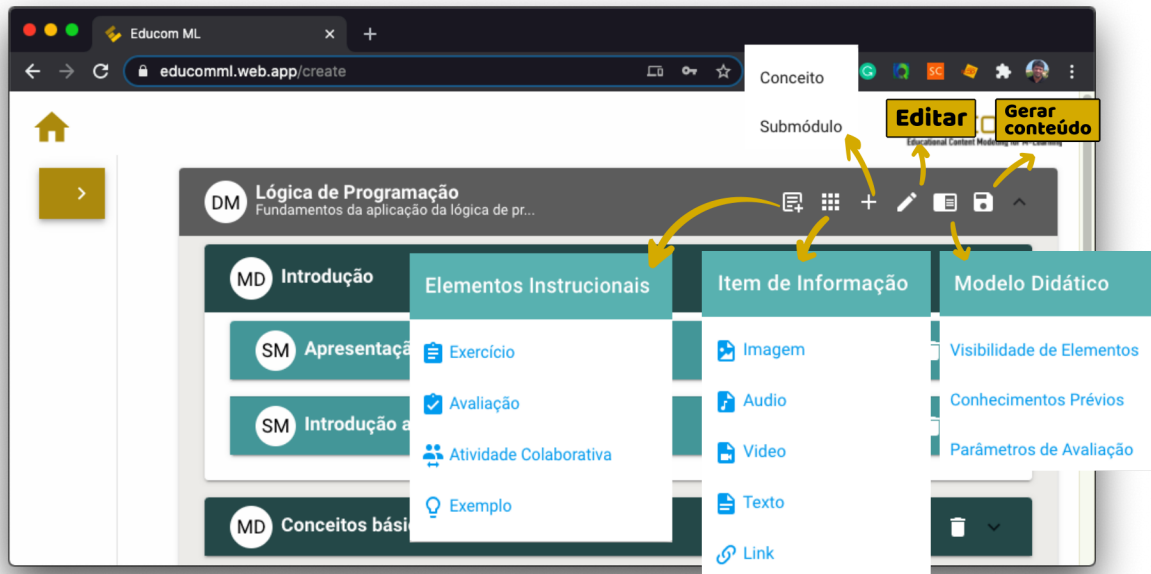
Fonte: elaborada pelo autor.

A Figura 25 apresenta a interface de acesso a definição de Módulos e Conceitos. Nessa interface também fica disponível o acesso a definição de Itens de Informação e Elementos Instrucionais. A seguir as funcionalidades presentes nessa interface são apresentadas:



Definição de módulo e conceito: esta opção apresenta ao professor um menu que possibilita criar um módulo ou conceito. Quando esta opção é apresentada na interface de modelagem de módulos o professor pode criar submódulos ou conceitos. Ao estabelecer os módulos e conceitos do domínio o professor está construído o Modelo Conceitual proposto na EDUCOM.ML.

Figura 25 – Interface para definição de módulos, conceitos, itens de informação, elementos instrucionais e modelo didático de um domínio.



Fonte: elaborada pelo autor.



Elementos Instrucionais: esta opção permite que o professor atribua “Exercícios, Exemplos, Avaliações e Atividades Colaborativas” a um domínio, módulo ou conceito. Ao definir os Elementos Instrucionais o professor está construindo o Modelo Instrucional proposto pela EDUCOM.ML.



Itens de Informação: esta opção permite que o professor defina Itens de Informação para os conceitos. Lembrando que Itens de Informação são representados com uso de mídia (textos, áudios, imagens, vídeos e links). Na verdade, na abordagem EDUCOM.ML Itens de Informação podem ser categorizados como princípio, conceito, fato e procedimento. No entanto, essas categorias são definidas de forma concreta pelo uso de mídias.



Edição: esta opção permite que o professor altere dados de entrada de módulos, conceitos, elementos instrucionais e itens de informação.



Definir modelo didático: esta opção permite que o professor defina os conceitos que serão apresentados aos aprendizes. Além disso, é nesta opção que são definidos os conhecimentos prévios de conceitos e os parâmetros que serão utilizados na avaliação da aprendizagem.



Gerar conteúdo: esta opção permite que o professor gere o conteúdo educacional do domínio modelado. Logo, ao selecionar essa opção instâncias do Modelo Educom (Conceitual, Instrucional e Didático), do Modelo Mobile e do Modelo React-native são construídas.

O protótipo EDUCOM TOOL foi desenvolvido com o *front-end Vue.js e JavaScript*. O *back-end* foi implementado com o *framework Django-Python*. Para manter a persistência de dados foram utilizados os banco de dados *PostgreSQL e Firebase*. No Capítulo 4 as interfaces de edição de módulos, conceitos, elementos instrucionais, itens de informação e do modelo didático são detalhadas na aplicação da abordagem EDUCOM.ML para o domínio de Lógica de Programação.

3.6 Considerações finais

Neste capítulo foi apresentado o conjunto de metamodelos que apoiam a abordagem de modelagem de conteúdo educacional para *m-learning*. Discutiu-se, ainda, o fluxo de construção de instâncias dos metamodelos, bem como qual o objetivo de cada um deles. A ideia principal é fornecer à equipe de produção de material didático um conjunto de modelos que orientam a construção de conteúdo educacional desconsiderando aspectos tecnológicos. Também, são fornecidos à equipe de desenvolvimento metamodelos para modelagem da plataforma móvel de desenvolvimento de uma aplicação educacional móvel.

Como discutido no início do capítulo, a EDUCOM.ML tem a proposta de ser uma ponte entre a equipe de desenvolvimento e a equipe de produção de conteúdo. A ponte criada pela EDUCOM.ML promove uma visão comum a todos interessados no projeto de modelagem de um domínio do conhecimento. Assim, tanto os responsáveis pela elaboração do conteúdo educacional, bem como os responsáveis pela implementação compartilham de uma visão unificada do projeto de construção de uma aplicação educacional móvel. Ademais, a EDUCOM.ML define as tarefas que são de responsabilidade de produtores de conteúdo e as tarefas atribuídas aos desenvolvedores de software. Essa divisão de tarefas proporciona à equipe de produção de conteúdo manter esforços na estruturação do conteúdo educacional, deixando assim aspectos técnicos sob a responsabilidade da equipe de desenvolvimento.

No próximo capítulo, a abordagem EDUCOM.ML é aplicada para modelar o conteúdo educacional de um domínio do conhecimento, bem como gerar o conteúdo para uma aplicação móvel. Além disso, o protótipo de aplicação web EDUCOM TOOL é utilizado para auxiliar o professor no processo de estruturação do conteúdo educacional.

APLICANDO A ABORDAGEM DE MODELAGEM DE CONTEÚDO EDUCACIONAL PARA M-LEARNING

Neste capítulo é apresentada uma aplicação da abordagem de modelagem EDUCOM.ML no domínio de Fundamentos de Lógica de Programação. O objetivo é demonstrar como cada modelo proposto na abordagem EDUCOM.ML é instanciado até que seja obtida a aplicação móvel com o conteúdo educacional.

A modelagem do conteúdo educacional envolveu o levantamento dos principais conceitos estudados na disciplina de Lógica de Programação do curso de Ciência da Computação da Universidade Estadual do Piauí. A partir da análise do conteúdo programático da disciplina, um conjunto de conceitos foram selecionados para a construção dos modelos propostos pela EDUCOM.ML. Além disso, foi utilizado o protótipo EDUCOM TOOL para a construção de instâncias dos modelos Conceitual, Instrucional e Didático. Este capítulo ainda apresenta interfaces da aplicação educacional móvel EDUCOMAPP gerada a partir de instâncias dos modelos Mobile e React-native da EDUCOM.ML.

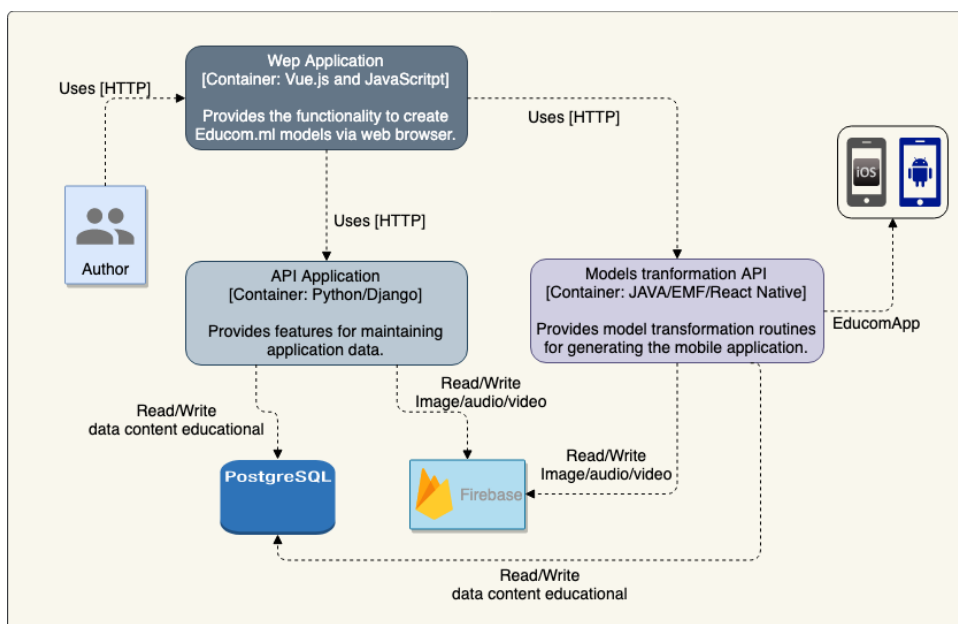
4.1 Modelagem do conteúdo educacional

Como apresentado no Capítulo 3, a abordagem de modelagem EDUCOM.ML propõe metamodelos para os modelos Conceitual, Instrucional, Didático, Mobile e React-native. Logo, a aplicação da abordagem EDUCOM.ML na construção do conteúdo educacional consiste na criação de instâncias de cada um desses modelos. O processo de construção dos modelos segue as etapas apresentadas na Figura 20 da Seção 3.1. Além disso, a aplicação EDUCOM TOOL foi utilizada para definição de módulos, conceitos, itens de informação e elementos instrucionais do domínio de Lógica de Programação.

A Figura 26 apresenta as tecnologias utilizadas no desenvolvimento do protótipo EDUCOM TOOL e na implementação de regras de transformação de modelos para obtenção do conteúdo educacional para dispositivos móveis. Em linhas gerais, tem-se a aplicação web EDUCOM TOOL desenvolvido com *Vue.js*, *JavaScript* e *PythonDjango* e para instanciar os metamodelos propostos pela EDUCOM.ML foi utilizado o *Eclipse Modeling Framework - EMF* (STEINBERG *et al.*, 2008) (ECLIPSE FOUNDATION, 2020). Com o objetivo de instanciar os modelos da EDUCOM.ML e implementar as regras de transformação de seus modelos, foi desenvolvida a *Models Transformation API*. A *Models Transformation API* é responsável por implementar as regras de transformação de modelo para modelo da EDUCOM.ML. Além disso, a *Models Transformation API* implementa regras de transformação de modelo para código que gera o conteúdo educacional modelado para uma aplicação móvel. Para manter a persistência de dados foram utilizados os bancos de dados *PostgreSQL* e *Firebase*. O *PostgreSQL* mantém dados referente à definição de módulos, conceitos, visibilidade de conteúdo educacional e do gerenciamento da aplicação, enquanto o *Firebase* destina-se especificamente à persistência de mídias como imagens, vídeos e áudios definidos na modelagem.

A aplicação da abordagem EDUCOM.ML no domínio da Lógica de Programação envolveu inicialmente a tarefa de especificar informações para a construção das instâncias dos modelos Conceitual, Instrucional e Didático. Essa tarefa foi executada com o auxílio do protótipo EDUCOM TOOL.

Figura 26 – Componentes de implementação de instância da Educom.ml.



Fonte: elaborada pelo autor.

4.1.1 Instanciando o Modelo Conceitual (MC)

Como destacado na Seção 3.3, o propósito do Modelo Conceitual é estabelecer uma hierarquia de conceitos que guiará a produção do conteúdo educacional. O processo de construção de uma instância do modelo conceitual é restrito à tarefa de identificação dos conceitos que serão modelados de um dado domínio, desconsiderando informações adicionais que expliquem cada um deles. Portanto, o primeiro passo no processo de construção do Modelo Conceitual é a segmentação do conteúdo do domínio do conhecimento em módulos. Na ferramenta EDUCOM TOOL essa tarefa é disponibilizada por um ícone representado pelo símbolo de “+”, como ilustra a Figura 27. Ao definir um módulo o autor precisa informar um título para o módulo e como informação opcional pode ser determinado um subtítulo. No contexto da modelagem do domínio de Lógica de Programação foram definidos 15 módulos.

Após a segmentação do conteúdo educacional em módulos, o passo seguinte no processo de construção do Modelo Conceitual é definir os conceitos a cada módulo. Na aplicação EDUCOM TOOL o autor tem acesso à definição de conceitos ao selecionar o ícone com símbolo de “+”, como ilustra a Figura 28. A definição de um conceito consiste em indicar o “nome do conceito”, “conhecimentos prévios” e “lista de relacionamento”.

Ao atribuir “conhecimentos prévios”, o autor estabelece quais conceitos o aprendiz precisa ter domínio para compreender o conceito atual. A definição de conhecimentos prévios

Figura 27 – Formulário de definição de módulos do domínio.



Fonte: Elaborada pelo autor.

Figura 28 – Formulário de definição de conceitos.

Edite o conceito selecionado.

Nome do conceito
Entrada de dados

16 / 100

CONHECIMENTOS PRÉVIOS

Lista de relacionamentos

Conceito	Nome da Relação	Tipo da Relação
Conceito de tipos de d...	ES-Tipo de dados	partOf

16 / 100

Defina as informações equivalentes aos conhecimentos prioritários desse conceito.

Conhecimentos prioritários +

Conceito Prévio
Conceito de tipos de dados

Identificador da prioridade
tipos de dados

Nível de prioridade
Obrigatório

Conceito Prévio
Tipo de dado literal

Identificador da prioridade
tipo literal

Nível de prioridade
Obrigatório

CANCELAR X SALVAR

CANCELAR X SALVAR

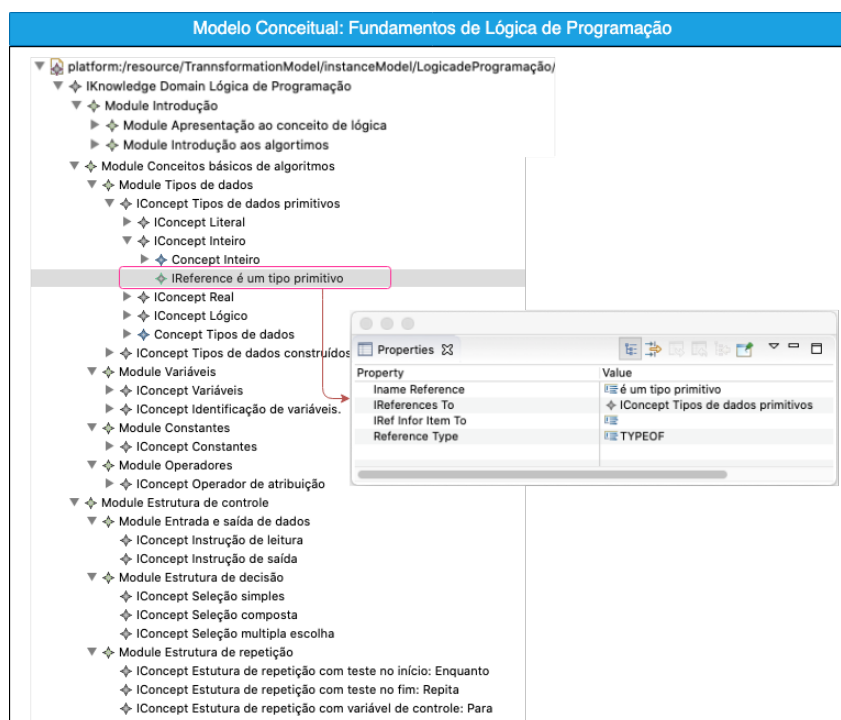
Fonte: Elaborada pelo autor.

permite que seja implementado regra de sequenciamento de estudo para o aprendiz, isto é, a aprendizagem é guiada conforme uma sequência de conceitos pré-estabelecidos. Estabelecer conhecimentos prévios é uma das ações desejável para a modelagem do Modelo Didático. Assim, ao definir conhecimentos prévios a um conceito o autor inicia o processo de construção do Modelo Didático. Observando um trecho da definição de conhecimentos prévios do conceito “Entrada de dados” na Figura 28, percebe-se que o aprendiz precisa estudar previamente os conceitos “tipos de dados” e “Tipo de dados literal”.

Com os módulos e conceitos do domínio de Lógica de Programação definidos na EDUCOM TOOL, um Modelo Conceitual foi instanciado. Essa instância do Modelo Conceitual foi obtida com a execução de regras de transformação que buscam informações na base de dados da EDUCOM TOOL e instancia um modelo conforme Metamodelo Educom (vide Figura 21). Como ilustração do Modelo Conceitual instanciado, a Figura 29 destaca trecho que apresenta a definição de um relacionamento estrutural entre o conceito *Inteiro* e *Tipos de dados primitivos*. Veja que a propriedade IREFERECSTO demonstra que foi estabelecida uma associação em que conceito *Inteiro* é um *Tipo de dados primitivo*. Logo, é possível perceber que o Modelo Conceitual instanciado reflete a modelagem especificada pelo autor na aplicação EDUCOM TOOL. Isto implica que qualquer ajuste realizado no conteúdo modelado é refletido no Modelo Conceitual instanciado com a execução das regras de transformação.

Vale lembrar que não existe um Modelo Conceitual único para um domínio do conhecimento. O fato é que aspectos como audiência, objetivo do ensino e aprendizagem, ambiente de destino do conteúdo e equipe de produção de conteúdo influenciam na modelagem do Modelo Conceitual. Logo, um domínio do conhecimento pode ser representado por diferentes modelos

Figura 29 – Atribuição de conceitos aos módulos de ensino.



Fonte: elaborada pelo autor.

conceituais.

4.1.2 Instanciando Modelo Instrucional (MI)

Conforme ilustrado na Figura 20 da Seção 3.3, após a modelagem do Modelo Conceitual o passo seguinte é a construção do Modelo Instrucional. Recuperando o propósito do Modelo Instrucional, tem-se que sua instância é composta de atribuições de informações complementares que ajudam o aprendiz a compreender cada conceito definido no Modelo Conceitual. Portanto, a tarefa desempenhada pelo autor na construção do MI é atribuir Itens de Informação (*conceito, princípio, fato e procedimento*) e Elementos Instrucionais (exemplos, exercícios, avaliações) a cada conceito modelado.

Como ilustra a Figura 25 da Seção 3.5, com a EDUCOM TOOL é possível definir Itens de Informação e Elementos Instrucionais. Itens de Informação e Elementos Instrucionais são representados concretamente por mídia *texto, imagem, áudio e vídeo*. A Figura 30 apresenta a definição de um Item de Informação para o conceito de “Tipos de dados” do domínio de Lógica de Programação. O formulário de definição de mídia texto fornece os seguintes campos:

- Classificação do item de informação.** Campo opcional que caracteriza se o texto representa um conceito, um fato ou princípio.
- Nível de dificuldade.** Permite determinar se o texto é para aprendiz iniciante ou avançado.
- Estilo de aprendizagem.** Permite determinar para qual estilo de aprendizagem (textual ou visual) o item de informação é mais adequado.
- Campo texto.**

Figura 30 – Formulário de definição de Item de Informação - mídia texto.

Fonte: Elaborada pelo autor.

Área para digitação do texto. Ressalta-se que apenas o campo de digitação de texto é obrigatório nesse formulário, porém quando os campos de definição de dificuldade e estilo de aprendizagem são informados pelo autor, cria-se as condições para a implementação de recursos de escolha de conteúdo conforme o perfil do aprendiz na aplicação desenvolvida.

Além de Item de Informação, foram atribuídos Elementos Instrucionais do domínio de Lógica de Programação. Nessa etapa, cada conceito pode receber informações que reforçam sua compreensão. Por exemplo, a Figura 31 apresenta o formulário de atribuição de exemplo para o conceito o “Tipos de dados”, sendo utilizado uma imagem para demonstrar quais são os valores considerados do tipo de dados primitivos. Importante destacar que Elementos Instrucionais também usam mídias para representar o conteúdo de Exercício, Avaliação, Atividade Colaborativa e Exemplo.

O processo de atribuição de Itens de Informação e Elementos Instrucionais foi aplicado a cada conceito especificado no Modelo Conceitual. Essas informações adicionais a cada conceito são refletidas em instâncias de classes que representam os Itens de Informação (II) e Elementos Instrucionais (EI) do Metamodelo Educom (Figura 21). A Figura 32 apresenta um trecho do Modelo Educom do domínio de Lógica de Programação instanciado com base na modelagem construída com a aplicação EDUCOM TOOL. O processo consiste em buscar na base de dados da EDUCOM TOOL informações para instanciar as classes do Metamodelo Educom. Como ilustra a Figura 32, o conceito “Tipo de dados” foi apresentado com um II::CONCEPT e sua representação

Figura 31 – Formulário de definição de Elemento Instrucional exemplo.

Fonte: Elaborada pelo autor.

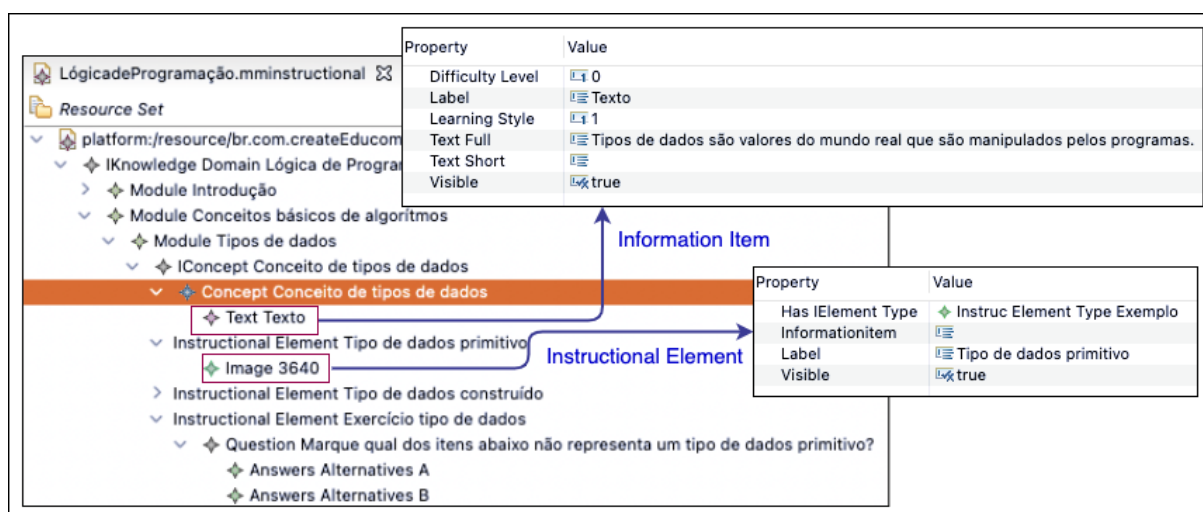
foi definida com um MOBILEMIDIA::TEXT. Para o mesmo conceito, foi definido EI::EXEMPLO com o uso de um MOBILEMEDIA::IMAGE.

O autor pode modelar diferentes Modelos Instrucionais para um dado domínio. Do mesmo modo do Modelo Conceitual, o objetivo do conteúdo modelado, a audiência e o meio de apresentação é que determinam quais Itens de Informação e Elementos Instrucionais serão definidos para os conceitos do domínio do conhecimento.

4.1.3 Instanciando Modelo Didático

Os modelos instanciados nas seções anteriores não estabelecem a sequência de estudo que o aprendiz deve seguir no conteúdo modelado. A definição do sequenciamento do conteúdo

Figura 32 – Trecho do Modelo Educom instanciado para Lógica de Programação.



Fonte: Elaborada pelo autor.

é especificada no Modelo Didático. Em resumo, uma instância do Modelo Didático pode conter informações que especificam parâmetros de avaliação (ASSESSMENTPARAMETERS), conhecimentos prévios (PRIORKNOWLEDGE) e seleção de conteúdo. Essas informações definem o roteiro de estudo do aprendiz.

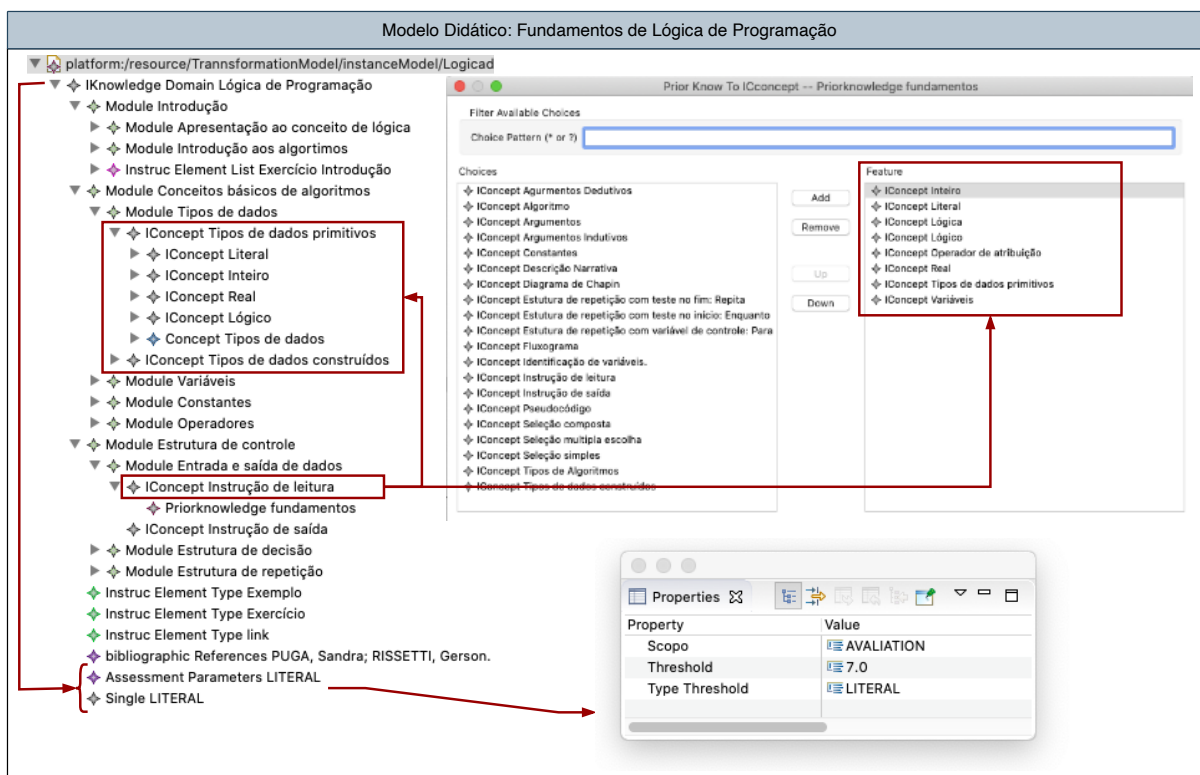
O parâmetro de avaliação define um valor que determina se o aprendiz alcançou o domínio de um dado conceito. Portanto, ao atingir o parâmetro de avaliação o aprendiz está apto a estudar um próximo módulo ou conceito. No domínio de Lógica de Programação, foi definido como parâmetro de avaliação o valor 7.0 para SINGLE::THRESHOLD com escopo específico de avaliação (SCOPO::AVALIATION). Isto implicar dizer que para considerar que o aprendiz atingiu um nível desejado de aprendizado, sua média nas avaliações realizadas precisam atingir um valor de acertos igual ou superior a 7.0.

Recuperando a Figura 21, um ASSESSMENTPARAMETERS pode ser representado por um único valor ou por faixa de valores. Além disso, pode ter um escopo de avaliação ou de domínio do conhecimento. Assim, quando o escopo é definido como o valor “avaliação”, implica que as avaliações determinam quando o aprendiz pode seguir para o próximo conceito ou módulo. No entanto, quando o escopo é definido como “domínio”, implica que o aprendiz só pode seguir para o próximo módulo ou conceito quando domina um determinado percentual do conteúdo. A Figura 33 ilustra o domínio de Lógica de Programação. Percebe-se em destaque que o parâmetro de avaliação tem um valor SINGLE::THRESHOLD igual a 7.0 e um escopo (SCOPO::AVALIATION) para avaliação. Nesse caso, as avaliações definidas para módulos e conceitos serão utilizadas para determinar se o aprendiz alcançou o nível de aprendizado desejado, podendo assim, seguir para o próximo módulo ou conceito. O autor pode definir parâmetros que especificam um escopo para todo o conteúdo educacional, ou então, pode definir pontualmente para cada módulo ou conceito.

Outro elemento do Modelo Didático representado pela classe PRIORKNOWLEDGE é a definição de conhecimentos prévios. A classe PRIORKNOWLEDGE mantém a relação de precedência entre os conceitos, isto é, quais conceitos anteriores o aprendiz precisa dominar para estudar um conceito. Logo, ao definir os conhecimentos prévios o autor está estabelecendo um guia de estudo para o aprendiz. Observando a Figura 33, percebe-se que antes do aprendiz estudar o conceito *Instrução de leitura* é preciso que os conceitos relacionados a *Tipos de dados primitivos* sejam estudados. Vale ressaltar que o Modelo Didático expressa um planejamento didático que orienta a implementação do conteúdo educacional. Em outras palavras, o Modelo Didático especifica requisitos de navegabilidade e parâmetros de avaliação que devem ser implementados pela equipe de desenvolvimento na construção de uma aplicação móvel.

A aplicação EDUCOM TOOL disponibiliza interface para a definição dos elementos que compõe o Modelo Didático. A Figura 34 apresenta um trecho da definição de conhecimentos prévios, parâmetro de avaliação e seleção de objetos modelados para Lógica de Programação. A interface 1 apresenta a lista de módulos definidos com um caixa de seleção para escolha

Figura 33 – Exemplo de Modelo Didático para o domínio de Lógica de Programação.



Fonte: elaborada pelo autor.

dos objetos que serão exibidos para o aprendiz no aplicativo móvel gerado. A definição de quais objetos modelados estarão presentes no conteúdo educacional é um recurso fornecido pelo Modelo Didático, cujo objetivo é permitir ao professor modelar o material didático conforme um objetivo do ensino e aprendizagem e da audiência que pretende-se alcançar. A interface 2 define os conhecimentos prévios de cada conceito modelado. Por fim, a interface 3 define os parâmetros de avaliação de módulos e conceitos.

Instanciados os modelos Conceitual, Instrucional e Didático do domínio de Lógica de Programação, tem-se uma estruturação do conteúdo educacional que se pretende ensinar. Importante destacar que esses modelos não detalham aspectos da tecnologia que será utilizada para apresentar o conteúdo educacional ao aprendiz, sendo necessário instanciar modelos que tratam desse contexto. Logo, com esse objetivo a EDUCOM.ML fornece metamodelos que tratam dos aspectos tecnológicos envolvidos na implementação de aplicativos educacionais móveis. Nas seções seguintes são apresentados os modelos específicos para implementação da plataforma alvo.

4.1.4 Instanciando Modelo Mobile

O Modelo Mobile foi projetado para representar uma abstração de composição de telas da aplicação móvel. A ideia do Modelo Mobile é proporcionar à equipe de desenvolvimento uma

Figura 34 – Interfaces de definição do Modelo Didático.

1 Defina as informações equivalentes ao modelo didático.

DESELECIONAR TUDO SELECIONAR TUDO

- ✓ Introdução
- ✓ Conceitos básicos de algoritmos
- ✓ Estrutura de controle
- ✓ Estrutura de dados
- ✓ Procedimentos e funções
- ✓ Estrutura de dados dinâmica

2 Defina as informações equivalentes aos conhecimentos prioritários desse conceito.

Conhecimentos prioritários +

Conceito Prévio	Identificador da prioridade	Nível de prioridade
Conceito de tipos de dados	tipos de dados	Obrigatório
Conceito Prévio	Identificador da prioridade	Nível de prioridade
Tipo de dado literal	tipo literal	Obrigatório

CANCELAR X SALVAR

3 Defina as informações equivalentes aos parâmetros de avaliação.

Parâmetros de avaliação +

Scopo	Elemento	Tipo de Threshold	Tipo de Dado
Avaliação	[MODULO] Conceitos ...	Literal	Simple

Valor: 7.0

CANCELAR X SALVAR

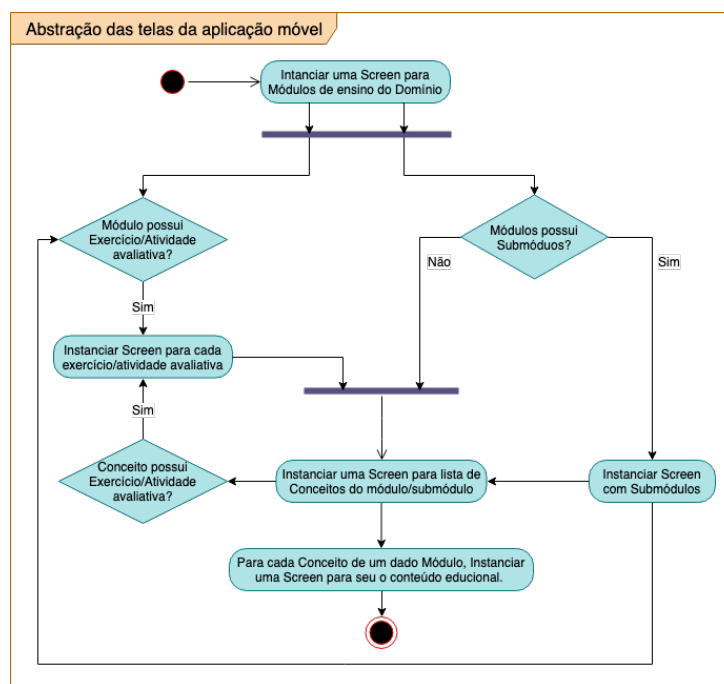
Fonte: Elaborada pelo autor.

visão da estrutura de telas criadas para apresentar o conteúdo educacional modelado. Importante ressaltar, que uma instância do Modelo Mobile é obtida sem a interferência de autores de conteúdo e da equipe de desenvolvimento, ou seja, o processo de geração do Modelo Mobile acontece de forma automática a partir da aplicação de regras de transformação modelo para modelos (Modelo Educom para Modelo Mobile). Contudo, alterações na instância gerada podem ser realizadas pela equipe de desenvolvimento para atender algum ajuste não implementado nas regras de transformação.

A Figura 35 apresenta um algoritmo em alto nível das regras de transformação do Modelo Educom em Modelo Mobile. Em linhas gerais, as regras de transformação implementam um algoritmo que cria uma tela para listar os módulos de um domínio do conhecimento. Caso um dado módulo tenha submódulos, uma nova tela é criada para listá-los. Caso um módulo tenha conceitos, uma nova tela é construída para apresentar os conceitos do módulo selecionado. Por fim, uma tela é criada para cada componente de mídia (texto, imagem, vídeo, áudio e link) definido para apresentar o conceito selecionado. Além de telas para módulos, conceitos e mídias, também são criadas telas para cada atividade avaliativa presente no Modelo Educom. Essas telas estão sempre associadas a um módulo ou conceito.

A Figura 36 ilustra um trecho das correspondências entre elementos do Modelo Educom e Modelo Mobile. Observa-se nos trechos destacados que a execução das regras de transformação do Modelo Instrucional em Modelo Mobile constrói um conjunto de telas (SCREEN) do aplicativo. Em destaque, podem ser observadas as telas criadas para o módulo “*Apresentação ao conceito de*

Figura 35 – Regra de transformação dos Modelos da Educom para Modelo Mobile.

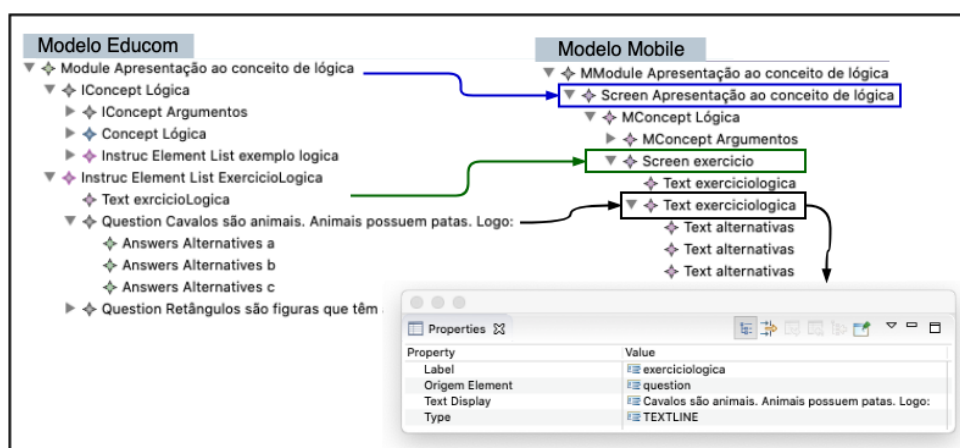


Fonte: elaborada pelo autor.

lógica” e para o “*exercício*” definido para o conceito “*Lógica*”. É possível ver ainda, detalhes das propriedades do elemento TEXT contido na tela (SCREEN) definida para o elemento instrucional “*exercício*”. Como já mencionado, esse processo de criação de telas para apresentar os objetos modelados no Modelo Educom prossegue até que todos os elementos do modelo de origem estejam representados no Modelo Mobile.

Para ilustrar como são implementadas as regras de transformação, a Listagem 1 apresenta o método que estabelece as correspondências entre elementos de atividade do Modelo Educom

Figura 36 – Trecho de correspondências entre elementos do Modelo Educom e Modelo Mobile.



Fonte: elaborada pelo autor.

com uma tela (SCREEN) do Modelo Mobile. Basicamente, o método recebe como parâmetro um elemento QUESTION do Modelo Educom e corresponde seus atributos com um elemento SCREEN e elementos de interface do Modelo Mobile, definindo assim como será apresentada na aplicação móvel a atividade.

Código-fonte 1 – Método de correspondência de questões entre Modelo Educom e Modelo Mobile

```

1 private Screen screenQuestion(Question question) {
2     Screen screenQuestion = mobileFactory.createScreen();
3     Text mText = mobileFactory.createText();
4     mText.setLabel(question.getDescriptionQuestion());
5     mText.setOrigemElement("Question");
6     mText.setTextDisplay(question.getOrderQuestion() + ") " + question.
7         getDescriptionQuestion());
8     int index = 1;
9     for (AnswersAlternatives iAnswer : question.getQuestionToAnswers()) {
10        Text mTextAnswers = mobileFactory.createText();
11        mTextAnswers.setLabel(iAnswer.getAnswers());
12        mTextAnswers.setTextDisplay(iAnswer.getIdObjAnswer() + ") " + iAnswer.
13            getAnswers());
14        mTextAnswers.setOrigemElement(mTextAnswers.getClass().getSimpleName());
15        mTextAnswers.setIsRightAnswer(iAnswer.getIsTrue());
16        if (!iAnswer.getIsTrue()) {
17            index++;
18        }
19        mText.getSubElementInterface().add(mTextAnswers);
20    }
21    screenQuestion.setLabel(mText.getLabel());
22    screenQuestion.setOrigemElement(question.getClass().getSimpleName());
23    screenQuestion.setTypeAvaliation(question.getTypeQuestion());
24    screenQuestion.getScreenToElementInterface().add(mText);
25    return screenQuestion;
26 }

```

Em resumo, o Modelo Mobile apresenta uma abstração do conjunto de telas que serão implementadas na plataforma móvel alvo. Destaca-se que nessa etapa aspectos relacionados à linguagem de programação para implementação da aplicação móvel são desconsiderados. Essa estratégia visa manter o Modelo Mobile independente, podendo assim ser implementadas regras de transformação para outras linguagens de programação. O passo seguinte nesse processo de geração do conteúdo educacional para uma aplicação móvel é a transformação do Modelo Mobile em um modelo de plataforma móvel alvo. Na seção seguinte um modelo de plataforma alvo é instanciado para o domínio de Lógica de programação.

4.1.5 Instanciando Modelo React-native (MRC)

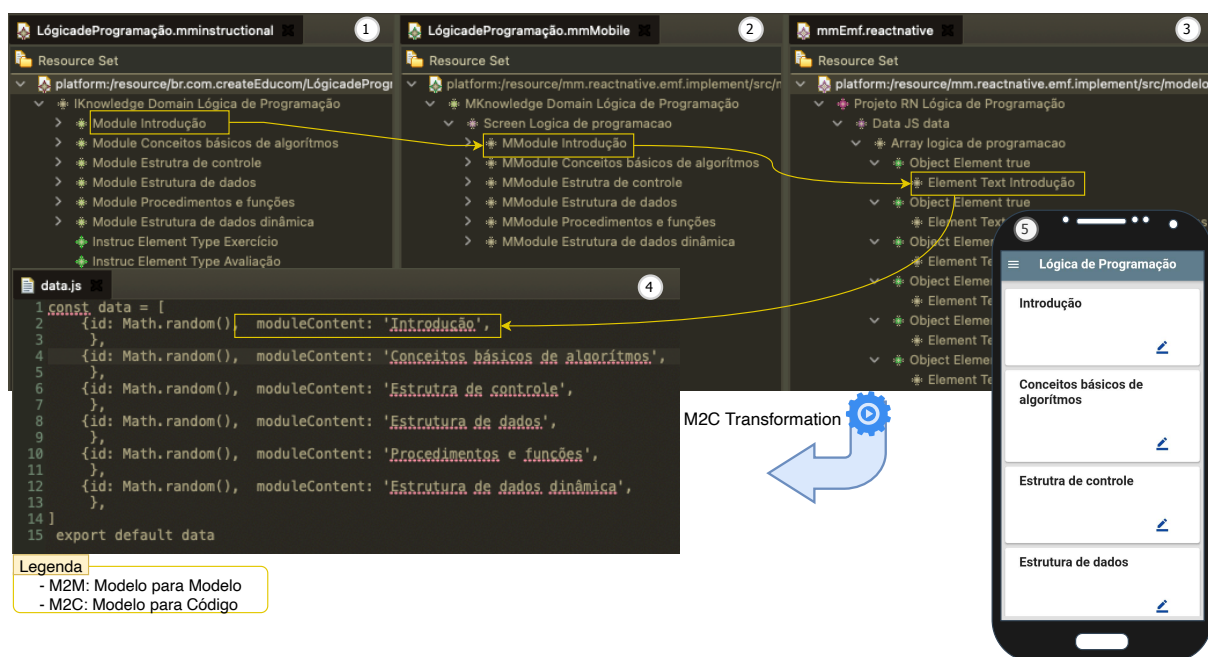
Nesta etapa, um modelo que representa as características da plataforma alvo é instanciando. Nesse modelo, aspectos como componentes de interface e estrutura de dados para armazenamento do conteúdo educacional são construídos. Na abordagem EDUCOM.ML o Metamodelo

React-native tem a finalidade de representar os componentes do *framework* de desenvolvimento de aplicações móveis React-native. Logo, todo conteúdo educacional modelado no Modelo Educom e no Modelo Mobile é correspondido em uma instância do metamodelo React-native.

A título de ilustração, a Figura 37 apresenta trechos dos modelos instanciados até a geração do Modelo React-native do domínio de Lógica de Programação. Nota-se que primeiro o Modelo Educom (ME) é instanciado e logo em um segundo momento regras de transformação (modelo para modelo) são aplicadas ao ME, gerando o Modelo Mobile (MM). Em um terceiro passo, são aplicadas regras de transformação (modelo para modelo) ao MM gerando o Modelo React-native (MR). Por fim, a aplicação de regras de transformação (modelo para código) geram o código-fonte React-native com conteúdo educacional que será apresentado no EDUCOMAPP.

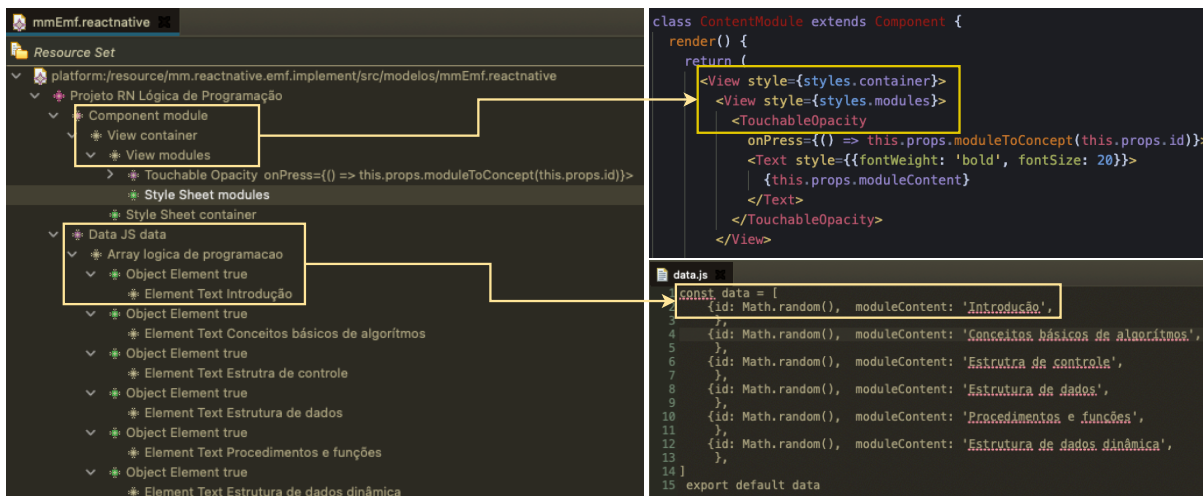
Esse processo de transformação dos modelos propostos pela abordagem EDUCOM.ML tem como resultado a implementação dos módulos, conceitos, itens de informação e elementos instrucionais definidos na modelagem do conteúdo educacional. A Figura 38 apresenta um trecho do Modelo React-native de Lógica de Programação e seu respectivo código-fonte obtido pela aplicação de regras de transformação de modelos. Fazendo uma leitura da figura, em amarelo, tem-se que o elemento “View” no Modelo React-native foi correspondido com a implementação de um componente correspondente em React-native. O mesmo aconteceu com o elemento “Data Js” do Modelo React-native, que gerou um arquivo “data.js” com o código-fonte que implementa uma estrutura de dados que disponibiliza o conteúdo educacional modelado para componentes da aplicação educacional móvel.

Figura 37 – Trecho dos modelos instanciados na abordagem de modelagem EDUCOM.ML.



Fonte: Elaborada pelo autor.

Figura 38 – Transformação do Modelo React-native em código-fonte React-native.



Fonte: Elaborada pelo autor.

O Modelo React-native é o último modelo instanciado para um conteúdo educacional modelado com a abordagem EDUCOM.ML. O código-fonte gerado a partir do MR é compilado criando a aplicação EDUCOMAPP. A seção a seguir apresenta a aplicação educacional móvel EDUCOMAPP.

4.1.6 Aplicativo móvel gerado

Nessa etapa, um protótipo de aplicação educacional móvel é gerado com o conteúdo educacional modelado com a EDUCOM.ML. Como apresentado na Figura 38, uma implementação React-native é obtida a partir do Modelo Mobile. Como exemplo, algumas imagens das telas de apresentação de módulos, conceitos e atividades de avaliação são ilustradas. O protótipo desenvolvido nessa etapa tem o objetivo de demonstrar a viabilidade da geração de conteúdo educacional e sua apresentação em aplicações educacionais móveis.

Inicialmente, tem-se a tela de abertura da aplicação EDUCOMAPP, como ilustra a Figura 39a. A tela seguinte apresentada a tela de acesso ao domínio modelado, como pode ser observado na Figura 39b. Cabe destacar que o protótipo foi implementado apenas com os recursos necessários para demonstrar que os metamodelos propostos na abordagem EDUCOM.ML são viáveis para geração de uma aplicação educacional móvel. Logo, recursos de controle de acesso ao aplicativo e de gerenciamento da aprendizagem serão implementados em uma versão final.

Ao selecionar a opção iniciar na tela de início, uma tela é apresentada com os módulos definidos para o domínio. A Figura 40a exemplifica o acesso ao conteúdo do “Conceito de tipos de dados”. Iniciando da primeira tela à esquerda, têm-se os módulos definidos para o domínio de Lógica de Programação. Ao selecionar “Conceitos básicos de algoritmo” uma nova tela é exibida com os conceitos definidos para o módulo. Por fim, as duas últimas telas apresentam ao aprendiz

Figura 39 – Telas iniciais do EDUCOMAPP.



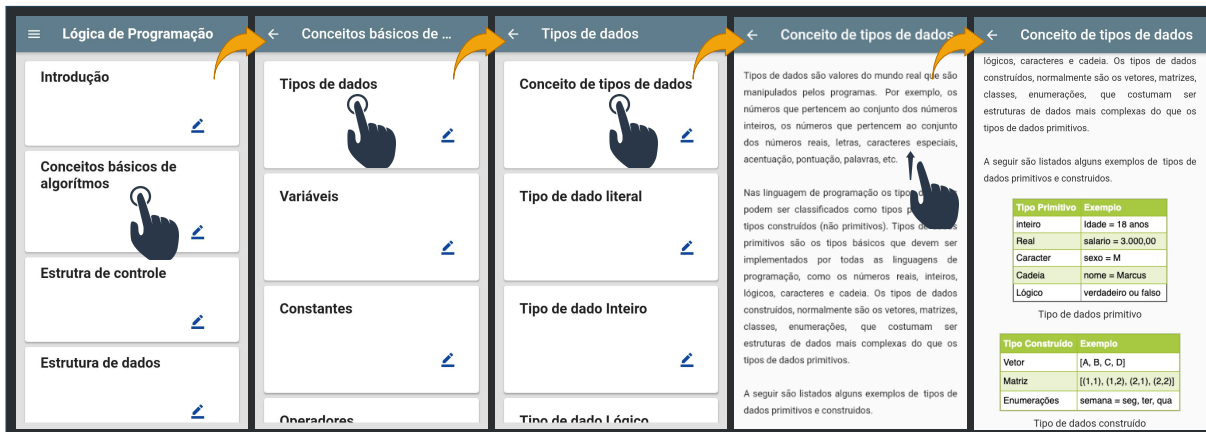
Fonte: Elaborada pelo autor.

o conteúdo educacional definido para “Conceito de tipos de dados”. É possível identificar que o “Conceito de tipos de dados” foi apresentado com uso de mídia texto e imagem.

Do mesmo modo, a Figura 40b apresenta o caminho para acesso a uma avaliação definida para o conceito “Raciocínio lógico”. Observe que a avaliação elaborada pelo autor preocupou-se em minimizar a entrada de dados por meio do uso do teclado. Isto é, ao elaborar uma atividade que permite que seja implementado recurso de *touchscreen* do dispositivo móvel, o autor contribuiu para que a equipe de desenvolvimento construísse uma interface que usou os recursos sensoriais dos dispositivos móveis. Essa é uma prática recomendável para a equipe de produção de conteúdo, pois minimiza a dificuldade de digitação em dispositivos móveis.

Em resumo, as interfaces geradas para o EDUCOMAPP refletem a abordagem de modelagem proposta pela EDUCOM.ML. O conteúdo apresentado ao aprendiz segue a estrutura de modularização dos conceitos estabelecidos no Modelo Conceitual, além da definição de Itens de Informação e Elementos Instrucionais determinadas no Modelo Instrucional. Além disso, as telas e os elementos que a compõe consideram a modelagem definida para o Modelo Mobile. Cabe ressaltar que EDUCOMAPP é um protótipo que se destina a demonstrar a viabilidade de geração de conteúdo educacional a partir da abordagem EDUCOM.ML. Por isso, uma versão final com recursos e funcionalidades que permitem o gerenciamento da aprendizagem será implementada posteriormente. Dentre as funcionalidades previstas para versão final estão a implementação do controle de sequenciamento do conteúdo modelado e avaliação do nível de aprendizagem do aprendiz. A implementação do recurso de sequenciamento de conteúdo considera os conhecimentos prévios definidos para cada conceito no Modelo Didático, enquanto

Figura 40 – Algumas interfaces do EDUCOMAPP para do domínio de Lógica de programação.



(a) Acesso ao conteúdo educacional modelado.



(b) Interface com exemplo de avaliação.

o nível de aprendizado do aprendiz considera os parâmetros de avaliação definidos no Modelo Didático.

4.1.7 Considerações finais

Neste capítulo, foi apresentada a aplicação da abordagem de modelagem de conteúdo educacional EDUCOM.ML. A abordagem foi aplicada ao domínio de Lógica de Programação do curso de Ciência da Computação da Universidade Estadual do Piauí. Para o domínio, foram instanciados os modelos Conceitual, Instrucional, Didático, Mobile e React-native. Além disso, para auxiliar o autor na produção do conteúdo educacional um protótipo de aplicação web foi implementando. A aplicação web EDUCOM TOOL possibilita que autores entrem com informações para a construção dos modelos Conceitual, Instrucional e Didático.

Além dos modelos que estruturam o conteúdo educacional, foram apresentados os modelos que representam a construção de um aplicativo móvel para exibição do domínio de Lógica de Programação ao aprendiz. Foram apresentadas instâncias do Modelo Mobile e

do Modelo React-native. O primeiro modela o conjunto de telas que apresentam o conteúdo educacional, enquanto o segundo modela a linguagem de programação que implementa o aplicativo móvel. Por fim, foram apresentadas algumas telas do aplicativo móvel gerado com o conteúdo educacional do domínio de lógica de programação.

No próximo capítulo é apresentada uma avaliação da EDUCOM.ML. A avaliação consistiu na utilização da abordagem de modelagem por autores, a fim de coletar o quanto a proposta de modelagem de conteúdo contribui para o processo de produção de material didático para dispositivos móveis.

AVALIAÇÃO DA EDUCOM.ML: PROVA DE CONCEITO COM O PROFESSOR

Neste capítulo, é apresentada a avaliação da proposta de estruturação de conteúdo educacional EDUCOM.ML. A avaliação tem como finalidade identificar, junto a professores, a percepção de utilidade dos modelos da EDUCOM.ML e das diretrizes de produção de mídia que apoiam o processo de produção de material didático.

Nas seções seguintes a condução da avaliação é detalhada. Na Seção 5.1 são descritos o planejamento, a metodologia e execução da avaliação. A Seção 5.2 apresenta a análise dos resultados obtidos a partir da avaliação conduzida. A Seção 5.3 trata dos riscos à validade da avaliação. Por fim, a Seção 5.4 apresenta as considerações finais do capítulo.

5.1 Planejamento, metodologia e execução

Para orientar a condução da avaliação de modo padronizado, foram utilizadas as diretrizes do método GQM (*Goal/Question/Metric*) (BASILI, 1993). O método GQM é uma abordagem orientada a objetivos para derivar métricas que contribuam para que os dados coletados sejam utilizáveis e atendam o propósito da avaliação. O escopo da avaliação da abordagem EDUCOM.ML pode ser resumido pelo GQM, com segue:

Analisar a proposta de modelagem EDUCOM.ML

Com o propósito de validar

Com respeito à utilidade percebida da abordagem EDUCOM.ML

Do ponto de vista do professor/autor

No contexto da modelagem de conteúdo educacional para ensino e aprendizagem móvel

5.1.1 Formulação das hipóteses

Com a definição do escopo da avaliação, uma questão de pesquisa foi formalizada considerando a seguinte hipótese:

- A utilização da abordagem de modelagem EDUCOM.ML ajuda/apoia a produção de conteúdo educacional para aprendizagem móvel?
 - H_0 : A Educom.ml não ajuda/apoia o professor no processo de produção de conteúdo educacional para aprendizagem móvel.
 - H_1 : A Educom.ml ajuda/apoia o professor no processo de produção de conteúdo educacional para aprendizagem móvel.

Para orientar a avaliação das hipóteses apresentadas, foram consideradas as seguintes variáveis:

- Variáveis Dependentes: percepção de utilidade dos participantes.
- Variáveis Independentes: material elaborado para apresentação da EDUCOM.ML; experiência dos participantes; experiência com produção de conteúdo educacional (móvel/tradicional).

A avaliação da abordagem de modelagem EDUCOM.ML foi realizada com a participação de 13 professores. Entre os participantes existiam docentes do ensino fundamental, médio e superior da rede particular e pública de ensino.

5.1.2 Instrumentação

A instrumentação para condução da avaliação com professores foi desenvolvida por meio dos seguintes artefatos:

- Uma página web¹ com um conjunto de vídeos que apresentam os modelos da EDUCOM.ML responsáveis pela estruturação do conteúdo educacional para aplicações educacionais móveis. Esses vídeos orientam o participante na construção dos modelos Conceitual, Instrucional e Didático, além de apresentar as diretrizes propostas para produção de texto, imagem e vídeo. Os participantes tiveram acesso a esses vídeos em um momento anterior à avaliação.
- Questionários online com itens que caracterizam os participantes² e que coletam aspectos sobre a avaliação³ da abordagem EDUCOM.ML. O questionário de perfil coletou

¹ Página web que apresenta a EDUCOM.ML e permite acesso à ferramenta web EDUCOM TOOL (www.educomml.web.app).

² Acesso ao questionário perfil do participante ([perfil](#)).

³ Acesso ao questionário de avaliação da Educom.ml ([avaliação](#)).

informações relacionado à experiência do participante, enquanto o questionário de avaliação coletou informações sobre a percepção dos participantes em relação ao apoio que abordagem EDUCOM.ML oferece no processo de produção do conteúdo educacional.

- Um protótipo de ferramenta web de autoria foi desenvolvido para ser utilizado como recurso de entrada do conteúdo educacional de um domínio do conhecimento. A ferramenta EDUCOM TOOL foi apresentada aos participantes durante a execução da avaliação da abordagem de modelagem EDUCOM.ML.
- Uma apresentação foi elaborada para o detalhamento da abordagem de modelagem EDUCOM.ML aos participantes.

5.1.3 Execução da avaliação

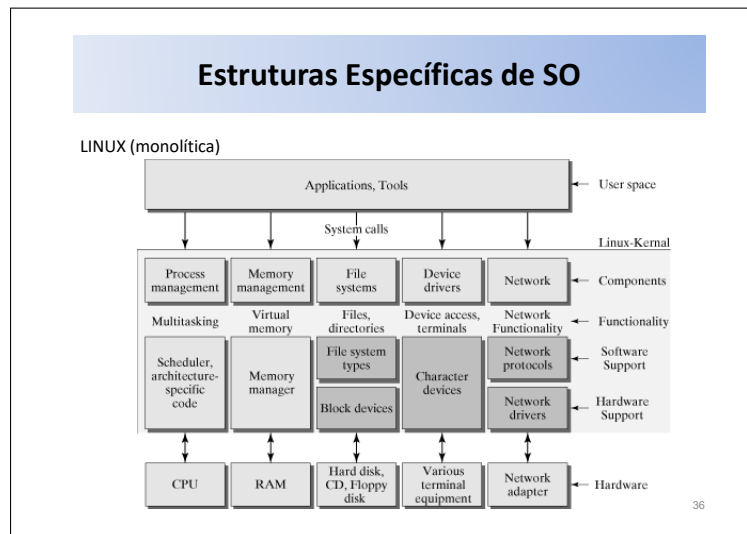
O procedimento de avaliação ocorreu por meio de encontros remotos (plataforma *Google Meet*) individualizados com os participantes. Vale esclarecer que a estratégia de encontros remotos deu-se para atender a imposição de distanciamento social exigido no enfrentamento da pandemia de COVID-19.

O primeiro momento da avaliação consistiu da apresentação da proposta de abordagem de modelagem EDUCOM.ML. Na apresentação, os conceitos que apoiam a construção dos Modelos Conceitual, Instrucional e Didático da EDUCOM.ML foram expostos aos participantes. Além dos modelos, foram ainda apresentadas as diretrizes para produção de mídias destinadas ao uso em aplicações educacionais móveis.

O segundo momento da avaliação consistiu da análise de material didático fornecido pelo participante. Previamente foi solicitado aos professores que encaminhassem material didático elaborado para ser ministrado presencialmente aos seus alunos. Assim, cada participante enviou slides utilizados anteriormente em cursos ministrados por eles. O objetivo dessa etapa era avaliar quais modificações seriam necessárias no material didático para se adequar a EDUCOM.ML. A Figura 41 ilustra um exemplo do material didático entregue pelo participante para ser avaliado. Observa-se que o slide apresenta uma imagem adequada a sala de aula, porém não oferece boa legibilidade para uso em dispositivos móveis. A imagem apresenta muita informação para ser exibida em equipamento com restrição de espaço em tela. A EDUCOM.ML sugere que imagens sejam produzidas observando a estética, relevância e dimensões adequadas aos dispositivos móveis. Essas diretrizes visam alertar professores que imagens produzidas precisam tornar a compreensão fácil e imediata, minimizando a carga cognitiva exigida do aprendiz.

O terceiro momento da avaliação foi demonstrar ao participante o uso da ferramenta de autoria de conteúdo educacional EDUCOM TOOL. A EDUCOM TOOL é um protótipo de aplicação web desenvolvido segundo a proposta de modelagem da EDUCOM.ML. A demonstração consistia em utilizar uma parte do material didático disponibilizado pelo participante para gerar o conteúdo educacional apresentado em uma aplicação educacional móvel. Portanto, o material didático

Figura 41 – Exemplo de material didático (slide) analisado na atividade de avaliação da EDUCOM.ML.



Fonte: Elaborada pelo autor.

fornecido pelo participante foi incluído na EDUCOM TOOL estabelecendo as informações necessárias para criar as instâncias dos modelos Conceitual, Instrucional e Didático. Com a estruturação do conteúdo educacional na EDUCOM TOOL, foi então gerada uma aplicação educação móvel do domínio modelado. Vale ressaltar que todo o procedimento de inclusão do conteúdo educacional da EDUCOM TOOL, bem como a geração da aplicação móvel ocorreram durante a avaliação.

Finalizando o processo de avaliação, foi solicitado aos participantes que respondessem o questionário de caracterização de perfil, bem como respondessem o questionário de avaliação da percepção de utilidade da abordagem EDUCOM.ML.

5.2 Análise e interpretação de resultados

Nesta etapa, os dados coletados são analisados e interpretados. Como técnica de análise de dados, foi utilizada a análise descritiva, que tem como ferramenta métodos de estatística descritivas para organizar, resumir e descrever aspectos importantes dos dados levantados em uma avaliação (REIS; REIS, 2002). As ferramentas descritivas adotadas nesse processo de análise de dados foram gráficos, tabelas e medidas de síntese, como porcentagens, índices e mediana.

Perfil dos participantes

O perfil dos participantes foi caracterizado com a aplicação de um questionário *online*. O questionário foi elaborado com itens relacionados à formação e experiência docente dos participantes. O objetivo é conhecer qual o grau de relação dos participantes com a adoção de

TICs na docência. Os dados coletados foram analisados por meio da construção de tabelas com frequência absoluta (Fa) e frequência relativa (Fr) das respostas. Além disso, gráficos foram plotados para apresentar os dados coletados, facilitando assim a tarefa de análise.

Uma das questões presente no questionário visava a caracterização dos participantes com relação a sua idade. Conforme ilustra a Tabela 14, a média de idade dos participantes é de 40 anos, com idades variando entre 28 e 52 anos. Assim, os dados sugerem maturidade no grupo de participantes selecionados para a avaliação.

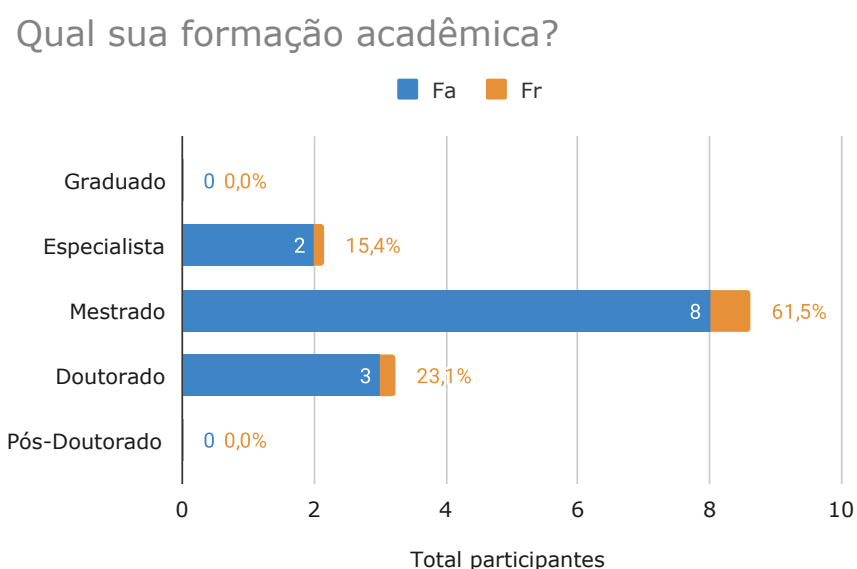
Tabela 14 – Perfil de idade dos participantes da avaliação.

Sujeito	Média	SD	Mediana	Min	Max
13	40	7	41	28	52

Fonte: Elaborada pelo autor.

Além de identificar a média de idade dos participantes, foram levantados informações sobre a formação e tempo de experiência docente. Em relação à formação docente, foi identificado que todos os participantes possuíam pós-graduação, com a seguintes distribuição: 08 mestres, 03 doutores e 02 especialistas. A Figura 42 sintetiza os dados que caracterizam a formação dos participantes.

Figura 42 – Distribuição de frequência de respostas sobre a formação acadêmica.

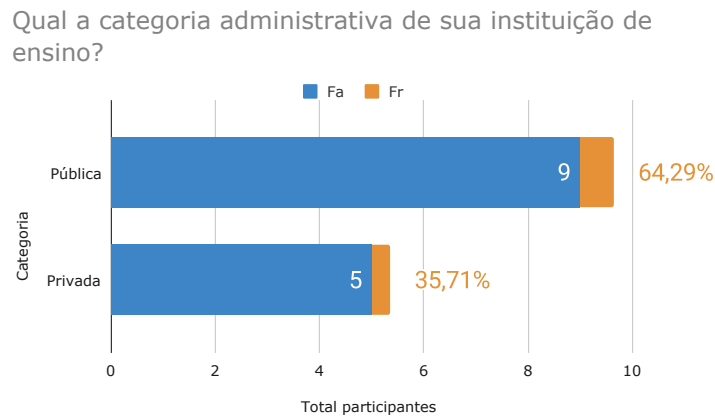


Fonte: Elaborada pelo autor.

Com relação à experiência docente dos participantes, foram questionados tempo de ensino, nível de ensino trabalhado e categoria de instituição que trabalha (pública/privada). Inicialmente, foi levantado junto aos participantes a categoria de entidade que lecionavam. A

Figura 43 apresenta a distribuição de participantes conforme a categoria de entidade de trabalho. Percebe-se que existe uma frequência dominante de 64,29% dos participantes trabalhando em instituições públicas, contra 35,72% em privadas. Além disso, dos 13 participantes, apenas 01 docente possuía experiência nas duas categorias de trabalho.

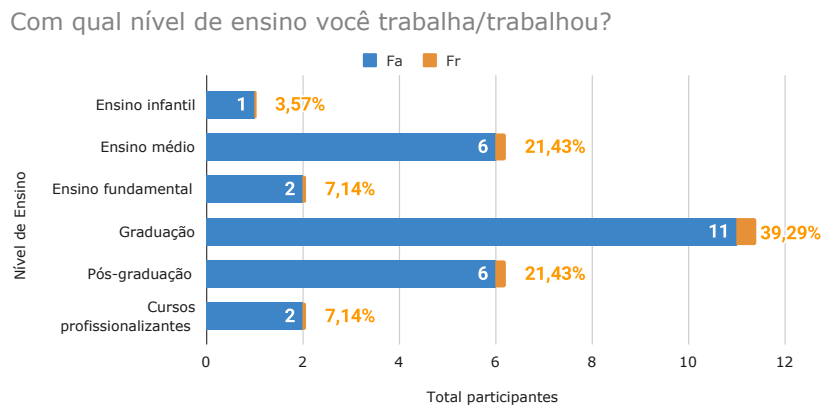
Figura 43 – Distribuição de frequência de respostas a categoria de entidade de trabalho.



Fonte: Elaborada pelo autor.

Outro ponto levantado no questionário de perfil consistiu em identificar em qual nível de ensino os participantes lecionavam. O objetivo deste questionamento é conhecer a experiência dos participantes em níveis de ensino diferentes. Como ilustra a Figura 44, a distribuição de frequência demonstra que os participantes atuam deste do ensino infantil a cursos profissionalizantes, com a predominância de 39,25% lecionando na graduação. Essa experiência diversificada dos participantes pode contribuir para uma avaliação mais abrangente com relação à produção de conteúdo educacional. Vale ressaltar que a divergência entre os totais de respostas e o número de participantes, ocorreu devido à existência de participantes que atuavam em mais de um nível de ensino.

Figura 44 – Distribuição de frequência de respostas a nível de ensino trabalhado.

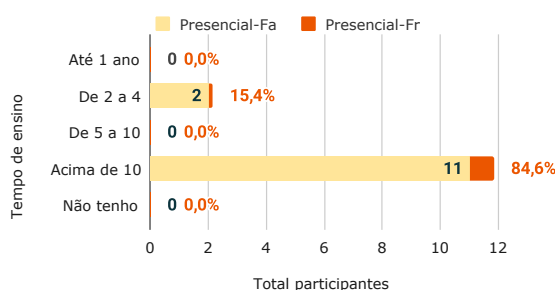


Fonte: Elaborada pelo autor.

A Figura 45 apresenta distribuição de respostas ao questionamento relacionado ao tempo de ensino na modalidade presencial. Como ilustra a Figura 45a, 84,6% dos participantes possuem experiência acima de 10 anos no ensino presencial. No entanto, quando foi questionada a experiência no ensino à distância, houve a predominância de participantes com experiência de até 01 ano, representando assim um percentual de 38,5%. Contudo, 30,8% dos participantes não possui nenhuma experiência no ensino à distância. A Figura 45b sintetiza esses dados, demonstrando que mesmo com um número de participantes reduzido nessa avaliação, os dados sobre a experiência docente sugerem que o ensino à distância possui uma adoção ainda tímida entre as modalidades de ensino relacionadas.

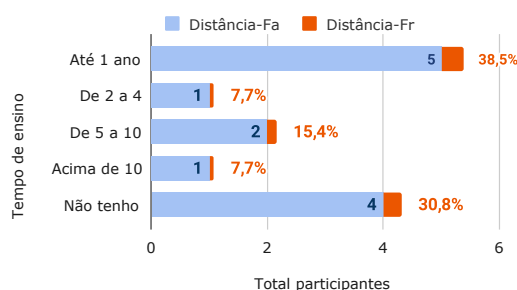
Figura 45 – Distribuição de frequência de respostas a tempo de experiência no ensino.

Indique qual o seu tempo de experiência no ensino presencial?



(a) Experiência ensino presencial.

Indique qual o seu tempo de experiência no ensino a distância?

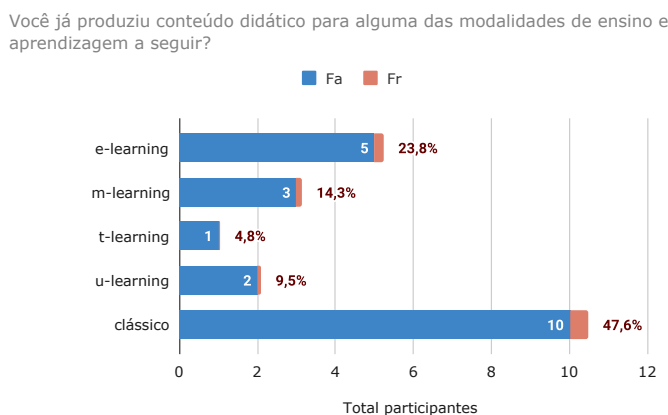


(b) Experiência ensino à distância.

No questionário de perfil também foi perguntado sobre a produção de conteúdo educacional para modalidades de ensino. O objetivo é identificar qual é a familiaridade do participante com a produção de material de didático para diferentes modalidades de ensino. Como apresenta a Figura 46, um percentual de 47,6% dos participantes produzem material para ensino clássico, enquanto 14,3% produzem conteúdo educacional para *m-learning*. Logo, considerando o público dessa avaliação, os dados levantados sugerem que *m-learning* é uma modalidade pouco trabalhada pelos participantes quando comparada com modalidade clássica e *e-learning*.

Uma questão relacionada ao uso de metodologias na produção de conteúdo educacional

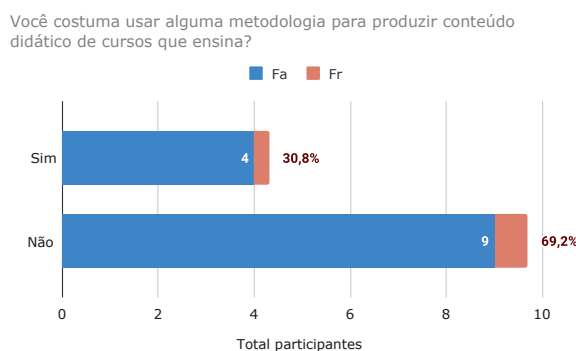
Figura 46 – Frequência de respostas sobre experiência em produção de conteúdo educacional.



Fonte: Elaborada pelo autor.

também foi incluída no questionário de perfil do participante. O intuito do questionamento é compreender se os participantes usam metodologias para elaboração de material didático. Como apresenta a Figura 47, 69,2% dos participantes não usam metodologia que oriente a produção do conteúdo educacional. Portanto, é sugestivo que propostas de modelagem possam contribuir no processo de elaboração de conteúdo educacional. Nesta questão, foi ofertado um campo descritivo no questionário para relacionar qual metodologia o participante fazia uso. Entre os participantes, dois informaram o uso de metodologia ativa e outro utilizou PBL - *Problem Based Learning*.

Figura 47 – Distribuição de frequência de respostas a uso de metodologia na produção de conteúdo educacional.

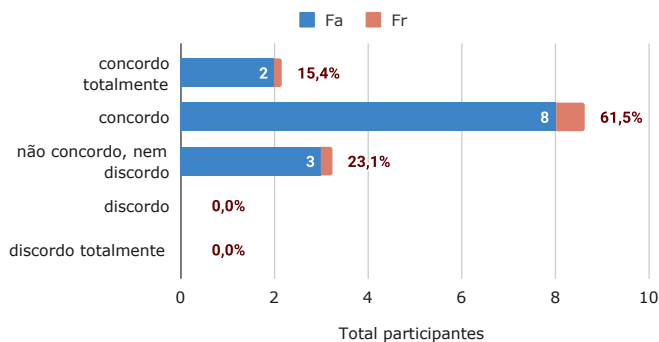


Fonte: Elaborada pelo autor.

Além de questionar a metodologia utilizada na produção de conteúdo educacional, foi perguntado sobre a facilidade de adaptação do material didático produzido a objetivos diferentes de ensino. Conforme ilustra a Figura 48, 61,5% dos participantes indicaram que produzem materiais didáticos fáceis de adaptação a diferentes objetivos e em um tempo razoável. Este questionamento foi realizado para obter uma visão de como a proposta de modelagem

Figura 48 – Distribuição de frequência de respostas a facilidade de adaptação de material didático.

Eu produzo material didático que são fáceis de serem adaptados a objetivos diferentes em um tempo razoável.



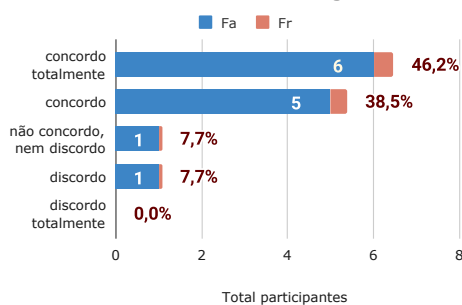
Fonte: Elaborada pelo autor.

EDUCOM.ML pode contribuir no processo de adaptação de conteúdo aos diferentes públicos e objetivos de ensino de forma ágil.

Com relação ao quanto familiar é o uso de tecnologias no processo de ensino para os participantes, dois questionamentos foram realizados. O primeiro tinha o objetivo de identificar o quanto os participantes apresentavam facilidade em trabalhar com a tecnologia. Para essa questão, a Figura 49a apresenta o resultado do questionamento, demonstrando que soma das respostas “concordo” e “concordo totalmente” resultou no percentual de 84,6% dos participantes, sugerindo que a relação com tecnologias não é um obstáculo para sua adoção no ensino. A Figura 49b apresenta o resultado do questionamento sobre o uso de TICs, pode-se perceber que o somatório de frequência de adoção de recursos tecnológicos é de 76,9% dos participantes. Novamente, o resultado sugere que a familiaridade com a tecnologia não é um impedimento para adoção de TICs.

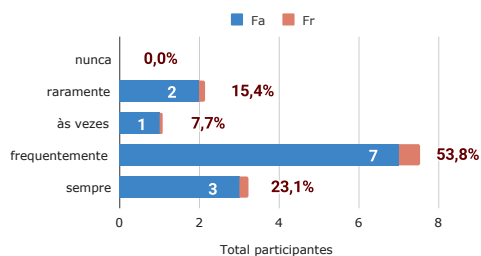
Figura 49 – Distribuição de frequência de respostas para familiaridade com TICs.

Eu tenho facilidade em utilizar tecnologias.



(a) Facilidade de uso de tecnologia.

Eu frequentemente utilizo tecnologias nos cursos que ensino (vídeos, podcasts, sites, plataformas EAD, entre outros).

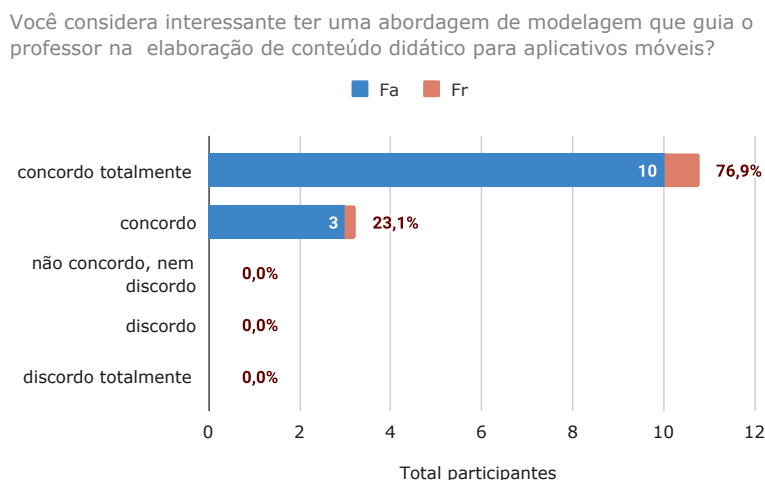


(b) Uso de TICs no ensino.

Fonte: Elaborada pelo autor.

Por fim, um questionamento relacionado com o interesse em ter uma abordagem de modelagem de conteúdo educacional para guiar a tarefa de produzir material didático para aplicativos educacionais móveis foi realizada aos participantes. Conforme demonstra a Figura 50, o somatório de respostas de concordância dos participantes foi de 100%, o que sugere que uma proposta de abordagem de modelagem para *m-learning* pode contribuir para adoção da aprendizagem móvel por professores.

Figura 50 – Distribuição de frequência de respostas sobre interesse em uma abordagem de modelagem para *m-learning*.



Fonte: Elaborada pelo autor.

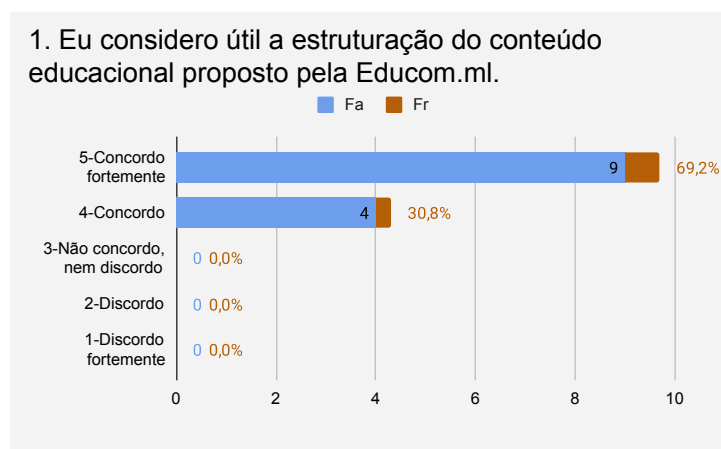
Avaliação da abordagem Educom.ml

A avaliação da abordagem EDUCOM.ML foi conduzida com a aplicação de um questionário com 07 itens aos participantes. Cada item do questionário apresentou uma escala *Likert* de 05 pontos, variando de “discordo totalmente” a “concordo totalmente”. O questionário foi elaborado com o objetivo de avaliar a percepção de utilidade das diretrizes de produção de mídia para aplicação educacionais móveis, bem como da estruturação de conteúdo educacional da EDUCOM.ML.

A primeira questão apresentada aos participantes estava relacionada à percepção de utilidade da estruturação do conteúdo educacional definido pela EDUCOM.ML. A questão consistia da seguinte afirmação: *Eu considero útil a estruturação do conteúdo educacional proposto pela EDUCOM.ML*. Como pode ser observado na Figura 51, a soma das opções “concordo e concordo fortemente” registrou o percentual de 100% dos participantes, sugerindo assim que a EDUCOM.ML pode ser um recurso útil no processo de estruturação do conteúdo educacional.

A segunda questão tinha como objetivo capturar a percepção dos professores com relação ao quanto a EDUCOM.ML ajuda no processo de produção de conteúdo educacional. A seguinte afirmação foi feita aos participantes: *Eu acredito que a abordagem de modelagem EDUCOM.ML*

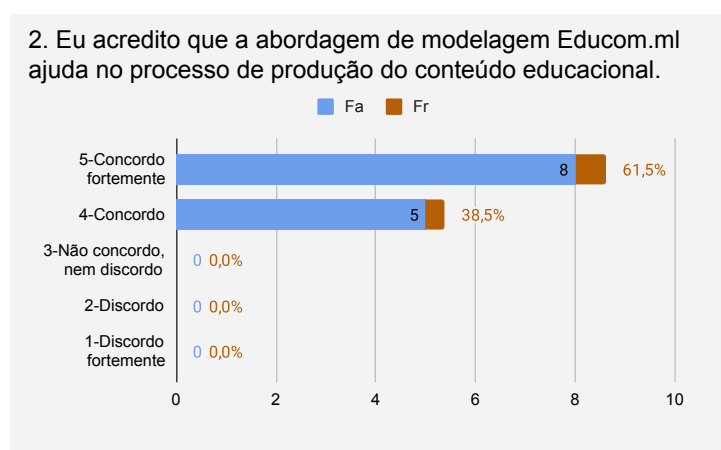
Figura 51 – Frequência de respostas sobre a utilidade da estruturação proposta pela EDUCOM.ML.



Fonte: Elaborada pelo autor.

ajuda no processo de produção do conteúdo educacional. A Figura 52 apresenta um gráfico que demonstra que os participantes da avaliação entendem que a proposta de abordagem de modelagem da EDUCOM.ML ajuda professores na produção de conteúdo educacional. O resultado apresenta um percentual de 100% dos participantes, quando as opções “concordo e concordo fortemente” são somadas. Esse resultado sugere a contribuição da EDUCOM.ML na produção de conteúdo educacional.

Figura 52 – Frequência de respostas sobre a utilidade da EDUCOM.ML no processo de produção de conteúdo.

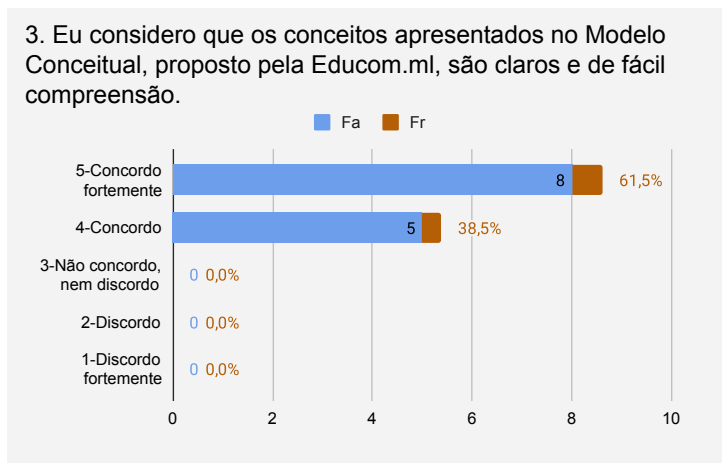


Fonte: Elaborada pelo autor.

A terceira pergunta do questionário de avaliação da EDUCOM.ML tinha como objetivo avaliar a percepção dos participantes quanto aos conceitos para construção do Modelo Conceitual. A Figura 53 apresenta o gráfico que resume o resultado obtido nesse item. Percebe-se que predominantemente as opções “concordo/concordo fortemente” foram selecionadas pelos participantes, resultando em 100% de concordância. O resultado sugere que os conceitos do

Modelo Conceitual são de fáceis entendimento.

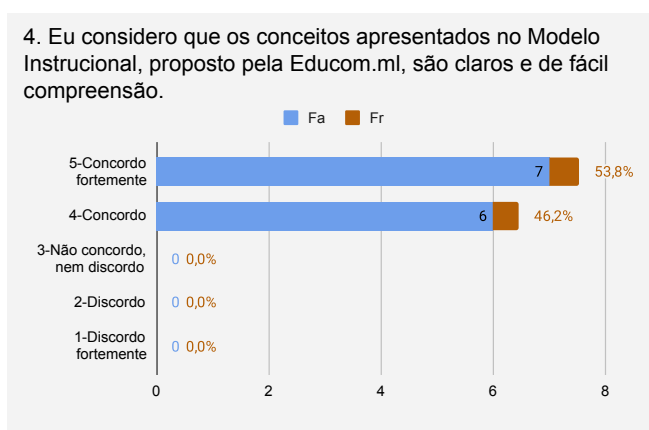
Figura 53 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Conceitual da abordagem EDUCOM.ML.



Fonte: Elaborada pelo autor.

A quarta questão da avaliação da EDUCOM.ML visa avaliar a percepção de facilidade de entendimento dos conceitos para construção do Modelo Instrucional. A Figura 54 apresenta o gráfico que resume o resultado obtido. Verifica-se no gráfico de frequência que as respostas predominantes dos participantes foram as opções “concordo/concordo fortemente”, sugerindo deste modo que os conceitos para construção do Modelo Instrucional são de fácil compreensão.

Figura 54 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Instrucional da abordagem EDUCOM.ML.

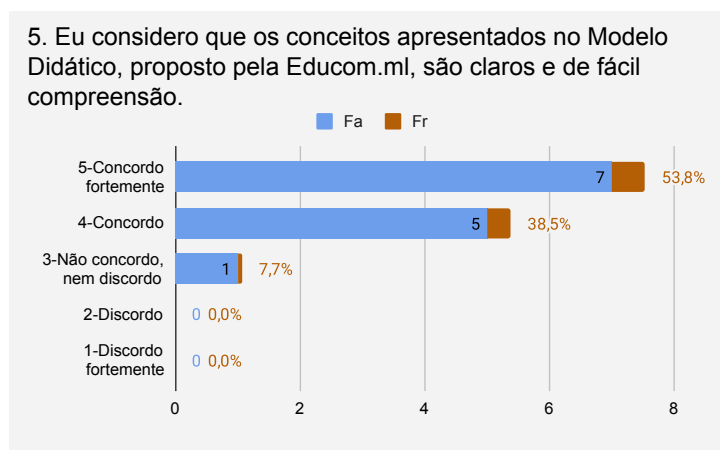


Fonte: Elaborada pelo autor.

A quinta questão da avaliação estava relacionada ao Modelo Didático da EDUCOM.ML. De modo similar às questões 3 e 4, foi solicitado aos participantes que respondessem se os conceitos para a construção do Modelo Didático eram de fácil compreensão. A Figura 55 apresenta o resultado desse questionamento. Nota-se que diferente das respostas para os modelos Conceitual

e Instrucional, nesse item houve a escolha da opção de neutralidade de um participante. Esse resultado chama atenção para que os conceitos do Modelo Didático sejam melhor esclarecidos.

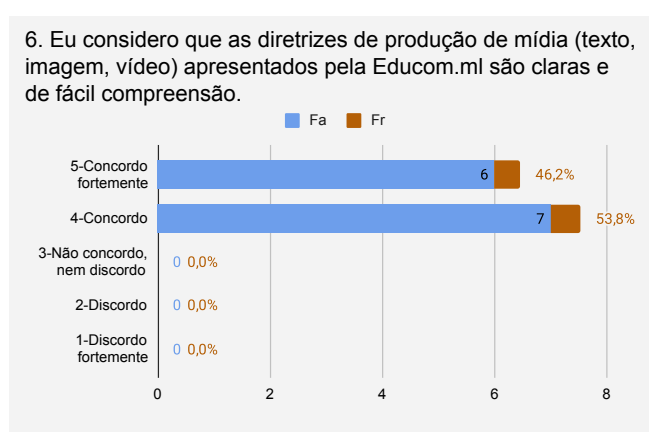
Figura 55 – Frequência de respostas sobre a facilidade de entendimento dos conceitos do Modelo Didático da abordagem EDUCOM.ML.



Fonte: Elaborada pelo autor.

A sexta questão trata da percepção de facilidade de compreensão das diretrizes de produção de mídia (texto, imagem e vídeo) da EDUCOM.ML. A Figura 56 apresenta o resultado, que demonstra a predominância de concordância (concordo/concordo fortemente) que representa 100% dos participantes. Assim, o resultado aponta para evidências de fácil compreensão das sugestões de produção de mídias presentes na Educom.ml.

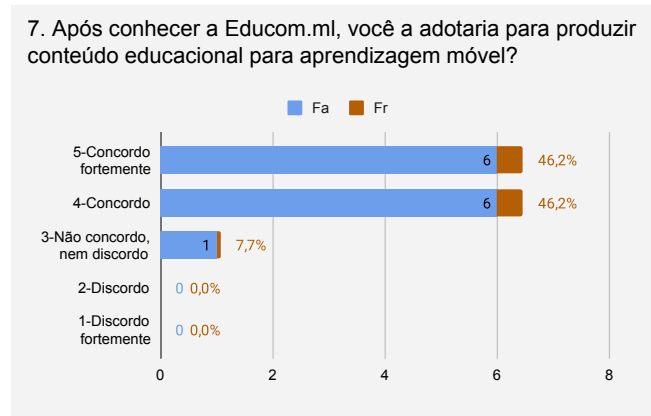
Figura 56 – Frequência de respostas sobre a compreensão das diretrizes de produção de mídia da abordagem EDUCOM.ML.



Fonte: Elaborada pelo autor.

A questão final da avaliação tinha o objetivo de avaliar a intenção dos participantes na adoção da EDUCOM.ML. A seguinte pergunta foi realizada aos participantes: *Após conhecer a Educom.ml, você a adotaria para produzir conteúdo educacional para aprendizagem móvel?*. A Figura 57 apresenta a frequência relativa de respostas a esse questionamento, sendo possível

Figura 57 – Frequência de respostas sobre o desejo de adoção da abordagem EDU-COM.ML no processo de produção de conteúdo educacional.

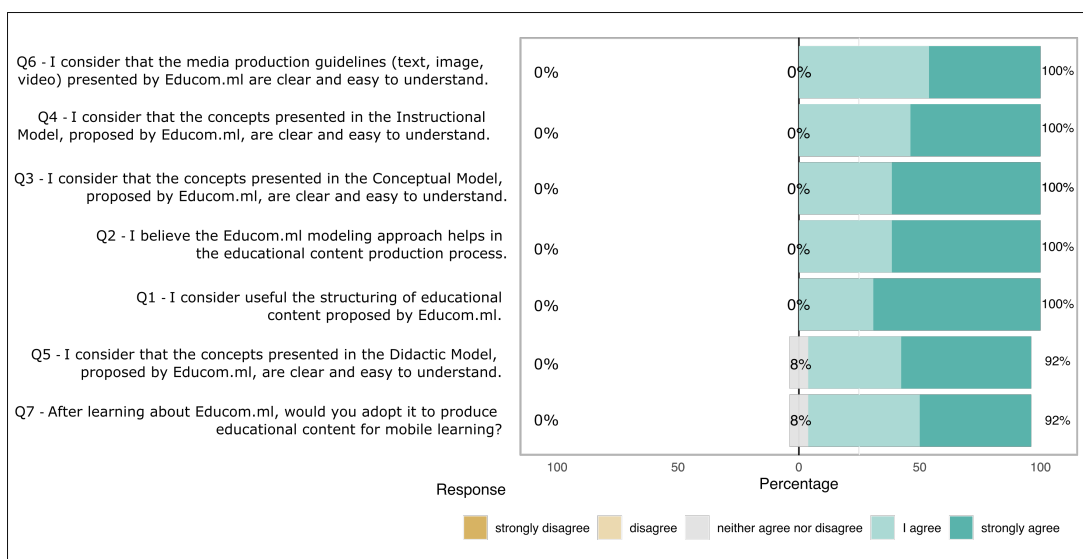


Fonte: Elaborada pelo autor.

notar que a soma das opções “concordo e concordo fortemente” atinge 100% dos participantes. Esse resultado sugere que a abordagem de modelagem EDUCOM.ML tem potencial de adoção na produção de conteúdo para aplicações educacionais móveis por professores.

O gráfico da Figura 58 apresenta um resumo dos resultados da avaliação da abordagem de modelagem Educom.ml. Note que entre as questões, nenhum professor respondeu com uma opção abaixo da escala neutra *Likert* utilizada. No gráfico, as questões Q1, Q2 e Q7 visam coletar a percepção de utilidade e possível adoção da abordagem EDUCOM.ML. Analisando as respostas às questões Q1 e Q2, a soma dos percentuais das opções “concordo” e “concordo totalmente” foi de 100%. Continuando a análise, para Q7, a soma dos percentuais das opções “concordo” e

Figura 58 – Visão geral da distribuição de frequência relativas de respostas da avaliação de utilidade da proposta de abordagem modelagem EDUCOM.ML.



Fonte: Elaborada pelo autor.

“concordo totalmente” foi de 92%. Assim, o resultado da avaliação aponta para evidências de que a abordagem EDUCOM.ML pode ser utilizada como ferramenta de apoio ao professor na produção de conteúdo educacional para aplicativo educacional móvel.

Para obter, de forma menos direcionada, a percepção de compreensão e utilidade dos modelos EDUCOM.ML, foram criados espaços para respostas abertas no questionário. Algumas das observações realizadas pelos professores participantes da avaliação estão listadas a seguir:

“Proporciona um direcionamento de conteúdo, auxiliando o docente na preparação de seus conteúdos.”

“A Educom.ml seria uma ferramenta que sistematizaria o trabalho docente, facilitando a preparação de um único conteúdo para diversas mídias.”

“A ideia é bastante promissora e vem suprir uma lacuna para muitos professores que estão só gravando os slides e repassando para os alunos.”

“As informações são claras, compreensíveis e ajudam no processo de produção de conteúdos *m-learning*. Porém, senti falta de informações mais específicas sobre configurações gráficas como sugestões para tamanho da fonte (ideal), dimensões do documento (melhor resolução para os slides, considerando a tela do celular), gravar vídeo com celular em pé ou deitado. Penso que isso deve ser considerado porque pode contribuir na qualidade do material produzido.”

“Observei que a estruturação proposta é absolutamente condizente com a realidade da esmagadora maioria dos meus alunos atuais.”

“Nem tenho como sugerir melhorias uma vez que o contato com a modelagem apresentada seguramente me ajudará em uma melhora na produção de minhas próprias aulas. Gostei muito do material!”

Assim como as questões com conteúdo direcionado, as respostas às perguntas abertas também apontaram para evidências de que os participantes consideraram EDUCOM.ML uma abordagem que apoia o professor na preparação de conteúdo educacional para aplicativos de aprendizagem móvel.

5.3 Riscos à Validade da Avaliação

Ao projetar uma avaliação é necessário identificar e entender os riscos que podem ocasionar distorções nos resultados obtidos a partir de sua execução. Nesta seção, destacam-se os riscos à validade interna, validade externa e de validade de conclusão.

Ameaça à validade de conclusão

A principal ameaça à validade de conclusão é o tamanho da amostra utilizada na avaliação. Para minimizar os efeitos da amostra reduzida, pretende-se em estudos futuros incrementar o quantitativo de participantes.

Riscos à validade interna

- **Compreensão da abordagem de modelagem:** devido à estratégia de encontros remotos, foi necessário elaborar um material adicional ao utilizado para apresentação da EDUCOM.ML. Foram gravados vídeos com a apresentação dos modelos da EDUCOM.ML, bem como das diretrizes de produção de mídia. Esses vídeos foram disponibilizados em uma página web com acesso livre.
- **Efeito fadiga:** a avaliação foi conduzida por encontros remotos com duração de uma hora e quinze minutos. Durante esse período, foram apresentados os conceitos da EDUCOM.ML, a ferramenta EDUCOM TOOL e respondidos os questionários de avaliação e perfil do participante. Nesse contexto, o tempo necessário para todo o processo de avaliação pode ter gerado fadiga aos praticantes.

Riscos à validade externa

- **Local do experimento:** devido ao momento de pandemia, a avaliação foi conduzida com encontros remotos. Essa estratégia dificultou a inclusão na avaliação de um momento destinado aos professores elaborarem material didático conforme a proposta de modelagem da EDUCOM.ML. Essa dificuldade será contornada em atividade de avaliação futura.
- **Número de participantes:** um número maior de participantes poderia retratar melhor a percepção de utilidade da abordagem EDUCOM.ML. Apesar de entender os riscos do número reduzido de participantes, foi possível obter indícios de que a proposta de modelagem de conteúdo educacional tem fundamentos que contribuem para elaboração de conteúdo didático mais adequado para aplicações educacionais móveis. Além disso, para fortalecer esses indícios, pretende-se planejar um experimento, em curto prazo, com uma quantidade mais expressiva de participantes. Esse experimento terá como alvo: i) professores, para avaliar a *Educom.ml* e elaborar conteúdo educacional para *m-learning*; ii) aprendizes para avaliar o conteúdo educacional produzido; iii) equipe de desenvolvimento de aplicativos móveis, para avaliar os metamodelos propostos pela EDUCOM.ML.

5.4 Considerações Finais

Neste capítulo foi apresentada uma avaliação da proposta de abordagem de modelagem de conteúdo educacional EDUCOM.ML. A avaliação teve como objetivo verificar a percepção de utilidade da EDUCOM.ML junto a professores produtores de conteúdo educacional.

A avaliação foi conduzida por encontros remotos com os participantes, sendo instrumentalizada com apresentação dos modelos Conceitual, Instrucional e Didático, além das diretrizes de produção de mídia estabelecida pela EDUCOM.ML. Com o objetivo de coletar a percepção de utilidade da EDUCOM.ML e da caracterização dos participantes foram aplicados os questionários de avaliação e perfil dos participantes. Os resultados da avaliação apresentam evidências que a abordagem de modelagem de conteúdo educacional EDUCOM.ML é uma proposta útil no processo de produção de conteúdo educacional para aplicações educacionais móveis.

No próximo capítulo será apresentada uma visão geral da pesquisa desenvolvida, juntamente com as principais contribuições deste trabalho de doutorado e suas limitações. Por fim, são apresentados os trabalhos futuros e as publicações associadas, resultantes deste trabalho.

CONCLUSÕES

Como abordado neste trabalho de doutorado, o avanço tecnológico cria oportunidades para o surgimento de novas modalidades de ensino e aprendizagem. Uma tecnologia emergente e que vem se tornando uma realidade para grande parte das pessoas é a tecnologia móvel. Para contextualizar o uso da tecnologia móvel atualmente, a *International Telecommunication Union - ITU* (UNION, 2020) realizou pesquisa que demonstra que de cada 100 habitantes da população global 75 possuem assinatura de banda larga móvel. Muito desse crescimento vem do menor custo de aquisição de dispositivos móveis em relação a computadores e de contrato de acesso à banda larga móvel. Além disso, dispositivos móveis apresentam constante crescimento na sua capacidade computacional. Essas características favorecem um cenário de fácil acesso à tecnologia móvel, estabelecendo assim um cenário oportuno para sua utilização no processo de ensino e aprendizagem.

Nesse contexto do avanço da tecnologia móvel, surge a aprendizagem móvel (*m-learning* — *mobile learning*), que se caracteriza pela adoção de dispositivos móveis como meio de acesso ao conteúdo educacional de um domínio do conhecimento. Além disso, a aprendizagem móvel traz o recurso da mobilidade no processo de ensino e aprendizagem, isto é, o aprendiz tem a escolha de determinar o local e o momento para a atividade de estudo. (SARRAB *et al.*, 2018)(JAHNKE; LIEBSCHER, 2020)(CHEN; THOMAS, 2020).

No cenário da tecnologia móvel, existem desafios de design e de implementação que precisam ser enfrentados no desenvolvimento de aplicações móveis (AMATYA; KURTI, 2013) (BIØRN-HANSEN; GRØNLI; GHINEA, 2018) (ASFOUR *et al.*, 2019). Portanto, esses desafios também estão presentes no desenvolvimento de aplicações educacionais móveis. Além disso, na construção de aplicações educacionais móveis existe a preocupação com a qualidade do conteúdo educacional produzido e de sua adequação a restrições (exemplo: tamanho de tela) e recursos (exemplo: touch, GPS, foto, vídeo, áudio) fornecidos por dispositivos móveis (CARVALHO; BARBOSA, 2018). Nesse sentido, a modelagem de conteúdo educacional é considerada uma

Figura 59 – Resumo de atividades desenvolvidas na pesquisa de doutorado.



Fonte: Elaborada pelo autor.

atividade relevante na estruturação do domínio que pretende-se ensinar (BARBOSA; BORGES; MALDONADO, 2013). Logo, neste trabalho foi proposto a EDUCOM.ML, que tem a finalidade de modelagem conteúdo educacional voltado para aprendizagem móvel.

Conforme discutido, o desenvolvimento de aplicação móvel é considerado complexo, pois requer que os desenvolvedores encontrem soluções que envolvem fatores como vasta oferta de modelos de dispositivos no mercado, a diversidade de sistemas operacionais existentes e as diversas linguagens de programação (nativa, híbrida, plataforma-cruzada) disponíveis para implementação da aplicação (BIØRN-HANSEN; GRØNLI; GHINEA, 2018). Nesse contexto, o trabalho de pesquisa propôs o uso da Engenharia Dirigida por Modelo para minimizar essas questões de diversidade de linguagens de programação disponíveis no mercado para o desenvolvimento de aplicações móveis. Ademais, a qualidade do conteúdo didático produzido é um fator importante para o aprendizado do aprendiz por meio de aplicativos móveis.

A Figura 59 sumariza as atividades de pesquisa conduzidas nesse trabalho de doutorado. A primeira atividade indica a motivação para o desenvolvimento da pesquisa. Portanto, duas questões foram investigadas: i) *Como modelar o conteúdo educacional no contexto de m-learning?*; e ii) *como ajudar o trabalho da equipe de produção de conteúdo e equipe de desenvolvimento no processo de desenvolvimento de aplicação educacional móvel?*

A segunda atividade no desenvolvimento da pesquisa foi propor um conjunto de adequações a abordagem de modelagem de conteúdo educacional AIM-CID para atender a modalidade de aprendizagem *m-learning*. Nesse perspectiva, foram propostos ajustes nos modelos Conceitual, Instrucional e Didático da AIM-CID, bem como um conjunto de diretrizes para produção de

mídias (texto, imagem e vídeo) para dispositivos móveis foi elaborado. Além disso, metamodelo para construção de instâncias concretas dos modelos da AIM-CID foi proposto, atendendo assim o processo de desenvolvimento de aplicação educacional móvel com a adoção da abordagem da Engenharia Dirigida por Modelos.

A terceira atividade é voltada para modelagem da plataforma alvo de desenvolvimento da aplicação educacional móvel. Portanto, metamodelos foram propostos para instanciar um modelo abstrato de aplicação educacional móvel, bem como foi proposto um metamodelo para a linguagem de desenvolvimento de aplicativos móveis React-native. Esta culmina na quarta atividade, que contempla a geração do protótipo EDUCOMAPP. O protótipo EDUCOMAPP foi construído para apresentar ao aprendiz o conteúdo educacional modelado pela EDUCOM.ML.

6.1 Contribuições da pesquisa

Este trabalho de doutorado teve como objetivo propor uma abordagem de modelagem para conteúdo educacional para aprendizagem móvel. O estudo desenvolvido buscou auxiliar professores e desenvolvedores na construção de aplicações educacionais móveis por meio da atualização da abordagem de modelagem de conteúdo AIM-CID e da abordagem de desenvolvimento de software MDE. Nesse contexto, a seguir as principais contribuições desta pesquisa são destacadas:

- Na linha de **Modelagem do Conteúdo Educacional**:
 - Atualização da abordagem AIM-CID (Abordagem Integrada de Modelagem — Conceitual, Instrucional e Didática) para contemplar a produção de conteúdo educacional considerando as limitações e recursos presentes em dispositivos móveis. Basicamente, os modelos Conceitual, Instrucional e Didáticos da AIM-CID foram adequados à estruturação do conteúdo educacional visando dispositivos móveis. Além disso, foi proposto um guia para orientar professores no processo de produção de mídias (texto, imagem e vídeo) para dispositivos móveis.
 - Proposição de um metamodelo para representar os modelos Conceitual, Instrucional e Didático da AIM-CID. Também foram adicionado ao metamodelo elementos para representar as mídias utilizadas na produção do conteúdo educacional.
 - Desenvolvimento do protótipo de aplicação web EDUCOM TOOL para ser utilizado como uma ferramenta de entrada do conteúdo educacional produzido conforme proposta de abordagem EDUCOM.ML. A partir do conteúdo educacional definido na EDUCOM TOOL, os modelos Conceitual, Instrucional e Didáticos são instanciados.
- Na linha de **Desenvolvimento de Aplicação Educacional Móvel**:

- Estabelecimento de uma abordagem da Engenharia Dirigida por Modelos para o desenvolvimento de aplicações educacionais móveis. A abordagem MDE proposta contempla dois momentos do desenvolvimento de aplicação educacional móvel. O primeiro está relacionado com a produção do conteúdo educacional que será utilizado na aplicação. Para esse primeiro momento, um conjunto de metamodelos foi definido para estruturar o conteúdo educacional. No segundo momento, a plataforma alvo de desenvolvimento da aplicação foi modelada. Para o segundo momento, metamodelos para representar uma aplicação móvel abstrata e a linguagem de programação de desenvolvimento foram propostos.
- Proposição do Metamodelo Mobile para representar uma abstração de telas que será implementada na aplicação educacional móvel.
- Proposição de regras de transformação de modelos para geração de um modelo abstrato de telas de uma aplicação móvel (instância do Modelo Mobile).
- Proposição de Metamodelo React Native para representar componentes da biblioteca javascript React-native. A aplicação educacional móvel gerada é implementada com a linguagem de programação React-native.
- Proposição de regras de transformação do Modelo Mobile e Modelo React-Native. Essas regras têm a finalidade de gerar um modelo React-Native com os requisitos de conteúdo educacional especificado no Modelo Mobile.
- Proposição de regras de transformação do Modelo React-Native em código-fonte React-native. O código-fonte gerado é responsável pela implementação da aplicação educacional móvel EDUCOMAPP.
- Desenvolvimento do protótipo funcional da aplicação educacional móvel EDUCOMAPP. A EDUCOMAPP tem o objetivo de apresentar o conteúdo educacional modelado com a abordagem EDUCOM.ML ao aprendiz.

6.2 Limitações da pesquisa

Ao fim do processo de pesquisa, foram identificados algumas limitações que precisam ser trabalhadas em um segundo momento. A seguir essas limitações são destacadas:

- **Avaliação da EDUCOM.ML:** a estratégia utilizada na avaliação da pesquisa sugeriu que a EDUCOM.ML pode contribuir para que professores adotem a EDUCOM.ML no processo de aprendizagem, porém, é necessário que experimentos com uma população maior de interessados sejam realizados. Portanto, novos experimentos devem ser planejados e conduzidos de modo a avaliar a abordagem EDUCOM.ML com professores, equipe de desenvolvimento e aprendizes, permitindo assim confirmar as evidências que a abordagem

de modelagem contribui para o processo de ensino e aprendizagem na modalidade *m-learning*.

- **Melhorar a ferramenta Educom Tool:** nesse estágio da pesquisa, a ferramenta EDUCOM TOOL teve como objetivo inicial fornecer uma interface de entrada do conteúdo educacional modelado pelo professor, permitindo assim a geração da aplicação educacional móvel. Portanto, recursos como formatação de texto não estão presentes nessa versão. Além disso, a ferramenta não contém recurso de ajuda para orientar o professor no design de mídia móvel conforme a proposta desta pesquisa.
- **Melhorias no aplicativo EducomApp:** o EducomApp é um aplicativo protótipo, com o objetivo de mostrar a viabilidade da construção de uma aplicação educacional móvel com a abordagem EDUCOM.ML. Portanto, recursos de acompanhamento do aprendiz no processo de aprendizagem não foram implementados. Esses recursos são necessários para orientar professores nas intervenções no conteúdo educacional modelado para atingir o objetivo da aprendizagem.

6.3 Trabalhos futuros

Lavando em consideração as limitações identificadas e visando dar continuidade à pesquisa conduzida no desenvolvimento do trabalho de doutorado, destacam-se como cenários para trabalhos futuros as seguintes questões:

- **Atualização da abordagem Educom:** a abordagem de modelagem pesquisada no doutorado teve como contexto a modalidade de aprendizagem móvel, que tem como suporte a tecnologia móvel e, por conseguinte o uso de dispositivos móveis como ferramenta de aprendizagem. A evolução dessa tecnologia é constante e, por isso, os novos recursos precisam ser incorporados à modelagem. Além disso, a EDUCOM é uma abordagem que trabalha com a MDE, portanto, novos metamodelos podem ser propostos para atender as características de outras modalidades de aprendizagem como, por exemplo, o uso de realidade aumentada, virtual e IoT. Além de atender as tecnologias emergentes, a abordagem EDUCOM pode propor metamodelos que apoiem recursos educacionais já consolidados como a geração de slides e documentos PDF.
- **Proposta de metamodelos para outras linguagens de desenvolvimento móveis:** além da biblioteca React-native a adotada na implementação da aplicação móvel gerada, outros metamodelos podem ser definidos para ampliar a variedade de linguagens de programação trabalhadas pela EDUCOM.ML.
- **Finalização da EDUCOM TOOL:** finalizar o protótipo EDUCOM TOOL com a inclusão de novos recursos como formatação de texto, análise de qualidade de imagem, orientação

de produção de mídia considerando a proposta de design da EDUCOM.ML. Além disso, implementar rotinas de recuperação de informação para produzir rascunho de conteúdo educacional a partir de conteúdos didáticos já produzidos pelo professor.

- **Finalização do protótipo de aplicação educacional móvel EDUCOMAPP:** finalizar a EDUCOMAPP com implementação de recursos como: i) gerenciamento do processo de aprendizagem; ii) controle de acesso à aplicação; iii) sequenciamento do conteúdo educacional conforme a definição de conhecimento prévios e parâmetros de avaliação estabelecidos no Modelo Didático da EDUCOM.ML; iv) gerenciamento de atividades colaborativas.
- **Planejamento e condução de novos experimentos:** planejar e executar novas avaliações da abordagem EDUCOM.ML do ponto de vista do professor, tutor, aprendiz e desenvolvedor de aplicações móveis. bem como planejar e executar experimentos para avaliar as versões finais da ferramenta de auditoria EDUCOM TOOL e a aplicação educacional móvel EDUCOMAPP.

6.4 Publicações

Como principais publicações resultantes das atividades conduzidas durante o presente trabalho de doutorado, podem-se destacar:

6.4.1 *Publicada*

- CARVALHO, Marcus Vinícius Ribeiro; BARBOSA, Ellen Francine; CARVALHO, Max Marcus C. **Educom.ml: A Modeling Approach for Mobile Educational Content**. In: 2020 IEEE Frontiers in Education Conference (FIE). IEEE, 2020. p. 1-8.
- CARVALHO, Marcus Vinícius Ribeiro; BARBOSA, Ellen Francine. **Uma abordagem de modelagem de conteúdo educacional móvel baseada na Engenharia Dirigida por Modelo**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2018. p. 338.

6.4.2 *Submetida*

- CARVALHO, Marcus Vinícius; BARBOSA, Ellen. **Structuring didactic content for mobile learning application: an MDE approach**. Informatics in Education, 2021.

REFERÊNCIAS

AHMAD, A.; LI, K.; FENG, C.; ASIM, S. M.; YOUSIF, A.; GE, S. An empirical study of investigating mobile applications development challenges. **IEEE Access**, IEEE, v. 6, p. 17711–17728, 2018. Citado na página 29.

AL-AMRI, S.; NOOR, N. F. B. M.; HAMID, S. B.; GANI, A. B. Designing mobile training content: Challenges and open issues. **IEEE Access**, IEEE, v. 8, p. 122314–122331, 2020. Citado na página 28.

ALALWAN, N.; ALZHRANI, A.; SARRAB, M. M-learning the next generation of education in cyberspace. In: WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY (WASET). **Proceedings of World Academy of Science, Engineering and Technology**. [S.l.], 2013. p. 642. Citado nas páginas 35 e 36.

ALBERTO, R. d. S. Model-driven engineering: A survey supported by the unified conceptual model. **Computer Languages, Systems & Structures**, v. 43, p. 139 – 155, 2015. ISSN 1477-8424. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1477842415000408>>. Citado na página 30.

AMATYA, S.; KURTI, A. Cross-platform mobile development: challenges and opportunities. In: SPRINGER. **International Conference on ICT Innovations**. [S.l.], 2013. p. 219–229. Citado na página 121.

APPLE. **Human Interface Guidelines**. developer.apple.com/ios, 2018. Available at <http://developer.apple.com/ios>. Accessed at 05/04/2018. Disponível em: <<https://developer.apple.com/ios>>. Citado nas páginas 59, 60, 61, 62, 63, 64 e 66.

ASFOUR, A.; ZAIN, S.; SALLEH, N.; GRUNDY, J. Exploring agile mobile app development in industrial contexts: A qualitative study. **International Journal of Technology in Education and Science**, International Journal of Technology in Education and Science, v. 3, n. 1, p. 29–46, 2019. Citado na página 121.

BANSAL, S. K.; DALRYMPLE, O. Imod-ont: Towards an ontology for instructional module design. In: **2016 IEEE Tenth International Conference on Semantic Computing (ICSC)**. [S.l.: s.n.], 2016. p. 354–357. Citado na página 28.

BARBOSA, E. F.; BORGES, V. A.; MALDONADO, J. C. Introdução à Modelagem de Conteúdos Educacionais. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 82–120, fev. 2013. ISSN 23167734. Disponível em: <<http://www.br-ie.org/pub/index.php/pie/article/view/2343>>. Citado nas páginas 27, 33, 35, 37, 38, 40, 46, 56 e 122.

BARBOSA, E. F.; MALDONADO, J. C. Mecanismos de Apoio à Modelagem de Conteúdos: Uma Contribuição ao Processo de Desenvolvimento de Módulos Educacionais. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 1, n. 1, p. 339–348, nov. 2004. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/335>>. Citado nas páginas 30, 37 e 50.

_____. An integrated content modeling approach for educational modules. In: _____. **Education for the 21st Century — Impact of ICT and Digital Resources: IFIP 19th World Computer Congress, TC-3, Education, August 21–24, 2006, Santiago, Chile**. Boston, MA: Springer US, 2006. p. 17–26. ISBN 978-0-387-34731-8. Disponível em: <http://dx.doi.org/10.1007/978-0-387-34731-8_3>. Citado nas páginas 27 e 30.

_____. Ima-cid: an integrated modeling approach for developing educational modules. **Journal of the Brazilian Computer Society**, v. 17, n. 4, p. 207–239, 2011. ISSN 1678-4804. Disponível em: <<http://dx.doi.org/10.1007/s13173-011-0043-5>>. Citado na página 46.

BASILI, V. R. Applying the goal/question/metric paradigm in the experience factory. **Software Quality Assurance and Measurement: A Worldwide Perspective**, London, UK: Chapman and Hall, v. 7, n. 4, p. 21–44, 1993. Citado na página 103.

BASSANI, J. **Uma abordagem baseada em Engenharia Dirigida por Modelos para Suportar o Teste de Sistemas de Software na Plataforma de Computação em Nuvem**. Dissertação (Mestrado) — Universidade Federal do Maranhão, 2012. Citado nas páginas 15, 43 e 44.

BÉZIVIN, J. Model driven engineering: An emerging technical space. In: LAMMEL, R.; SA-RAIVA, J.; VISSER, J. (Ed.). **Generative and Transformational Techniques in Software Engineering**. Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 4143). p. 36–64. ISBN 978-3-540-45778-7. Disponível em: <http://dx.doi.org/10.1007/11877028_2>. Citado nas páginas 30, 42 e 43.

BIARD, N.; COJEAN, S.; JAMET, E. Effects of segmentation and pacing on procedural learning by video. **Computers in Human Behavior**, 2017. ISSN 0747-5632. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563217306829>>. Citado nas páginas 27, 58, 59, 65 e 66.

BIØRN-HANSEN, A.; GRØNLI, T.-M.; GHINEA, G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 5, p. 1–34, 2018. Citado nas páginas 121 e 122.

BORGES, V. A. **Uma contribuição à modelagem e geração automática de conteúdos educacionais**. Dissertação (mathesis) — Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos,, 2010. Disponível em: <<http://dx.doi.org/10.11606/D.55.2010.tde-21062010-102043>>. Citado nas páginas 27, 28, 30, 33, 35 e 37.

BRAGA, J.; PIMENTEL, E.; DOTTA, S.; MENESES, L. Desafios sobre a avaliação dos ganhos de aprendizagem na educação mediada por tics. In: **Anais do DesafIE-III Workshop de Desafios da Computação Aplicada à Educação, CSBC**. [S.l.: s.n.], 2014. Citado na página 27.

BRAGA, J. C.; PIMENTEL, E.; DOTTA, S. Metodologia INTERA para o desenvolvimento de Objetos de Aprendizagem. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 24, n. 1, p. 306, 2013. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2509>>. Citado nas páginas 33, 37 e 49.

BUCCHIARONE, A.; CABOT, J.; PAIGE, R. F.; PIERANTONIO, A. Grand challenges in model-driven engineering: an analysis of the state of the research. **Software and Systems Modeling**, Springer, v. 19, n. 1, p. 5–13, 2020. Citado na página 30.

CARVALHO, M. V.; BARBOSA, E. F. Uma abordagem de modelagem de conteúdo educacional móvel baseada na engenharia dirigida por modelo. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2018. v. 29, n. 1, p. 338. Citado na página 121.

CARVALHO, M. V.; LOPES, D.; ABDELOUAHAB, Z. A framework based on model driven engineering to support schema merging in database systems. In: **New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering**. [S.l.]: Springer, 2015. p. 397–405. Citado na página 30.

CASTRILLO, M. D.; MARTÍN-MONJE, E.; BÁRCENA, E. Mobile-based chatting for meaning negotiation in foreign language learning. **International Association for Development of the Information Society**, ERIC, 2014. Citado na página 34.

CASTRO-ALONSO, J. C.; KONING, B. B. de; FIORELLA, L.; PAAS, F. Five strategies for optimizing instructional materials: Instructor-and learner-managed cognitive load. **Educational Psychology Review**, Springer, p. 1–29, 2021. Citado nas páginas 65 e 66.

CHAVOSHI, A.; HAMIDI, H. Social, individual, technological and pedagogical factors influencing mobile learning acceptance in higher education: A case from iran. **Telematics and Informatics**, v. 38, p. 133 – 165, 2019. ISSN 0736-5853. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S073658531830618X>>. Citado nas páginas 34 e 37.

CHEN, C.-M.; LIN, Y.-J. Effects of different text display types on reading comprehension, sustained attention and cognitive load in mobile reading contexts. **Interactive Learning Environments**, Taylor & Francis, v. 24, n. 3, p. 553–571, 2016. Citado nas páginas 58 e 59.

CHEN, H.-T. M.; THOMAS, M. Effects of lecture video styles on engagement and learning. **Educational Technology Research and Development**, Springer, p. 1–18, 2020. Citado na página 121.

CHIMALAKONDA, S.; NORI, K. V. Idont: An ontology based educational modeling framework for instructional design. In: **2013 IEEE 13th International Conference on Advanced Learning Technologies**. [S.l.: s.n.], 2013. p. 253–255. ISSN 2161-3761. Citado nas páginas 28 e 49.

CHITTARO, L. Distinctive aspects of mobile interaction and their implications for the design of multimodal interfaces. **Journal on Multimodal User Interfaces**, Springer, v. 3, n. 3, p. 157–165, 2010. Citado nas páginas 59, 60 e 66.

COMPANY, F. **Free Vectors, Stock Photos & PSD Downloads | Freepik**. 2020. <<https://www.freepik.com/>>. (Accessed on 11/18/2020). Citado nas páginas 63 e 65.

CURUM, B.; KHEDO, K. K. Cognitive load management in mobile learning systems: principles and theories. **Journal of Computers in Education**, Springer, p. 1–28, 2020. Citado nas páginas 27, 29 e 57.

DANISH, J.; HMELO-SILVER, C. E. On activities and affordances for mobile learning. **Contemporary Educational Psychology**, v. 60, p. 101829, 2020. ISSN 0361-476X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0361476X19304345>>. Citado nas páginas 27 e 34.

DIONISIO, C. S. **A Project Manager's Book of Forms: A Companion to the PMBOK Guide**. [S.l.]: John Wiley & Sons, 2017. Citado na página 50.

DIWANJI, P.; SIMON, B. P.; MARKI, M.; KORKUT, S.; DORNBERGER, R. Success factors of online learning videos. In: **2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2014)**. [S.l.: s.n.], 2014. p. 125–132. Citado nas páginas 58, 59, 65 e 66.

DUARTE, N. F. D. F.; BARBOSA, E. F. A requirements catalog for mobile learning environments. In: **Proceedings of the 28th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2013. (SAC '13), p. 1266–1271. ISBN 978-1-4503-1656-9. Disponível em: <<http://doi.acm.org.ez67.periodicos.capes.gov.br/10.1145/2480362.2480599>>. Citado nas páginas 36 e 37.

ECLIPSE FOUNDATION. **Eclipse Modeling Framework (EMF)**. 2020. Disponível em: <<https://www.eclipse.org/modeling/emf/>>. Acesso em: 20/01/2020. Citado na página 86.

EISENMAN, B. **Learning React Native**. Second. [S.l.]: O'Reilly Media, Inc., 2017. Citado na página 77.

ELIAS, T. 71. universal instructional design principles for mobile learning. **International Review of Research in Open and Distributed Learning**, Athabasca University Press (AU Press), v. 12, n. 2, p. 143–156, 2011. Citado na página 59.

ENNOUAMANI, S.; MAHANI, Z.; AKHARRAZ, L. A context-aware mobile learning system for adapting learning content and format of presentation: design, validation and evaluation. **Education and Information Technologies**, Springer, p. 1–37, 2020. Citado nas páginas 28, 29 e 37.

FACEBOOK. **Documentation React Native**. 2020. Library Catalog: reactnative.dev. Disponível em: <<https://www.reactnative.dev/>>. Citado na página 77.

FARJON, D.; SMITS, A.; VOOGT, J. Technology integration of pre-service teachers explained by attitudes and beliefs, competency, access, and experience. **Computers & Education**, Elsevier, v. 130, p. 81–93, 2019. Citado na página 29.

FILHO, N. F. D.; CRISTIANO, W. Oceanschool-sistema de aprendizagem móvel com foco no treinamento para a certificação oceb. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2016. v. 27, n. 1, p. 567. Citado nas páginas 30 e 37.

FU, Q.-K.; HWANG, G.-J. Trends in mobile technology-supported collaborative learning: A systematic review of journal publications from 2007 to 2016. **Computers & Education**, Elsevier, v. 119, p. 129–143, 2018. Citado na página 27.

GAMBOA, A. X. R.; DURAN, D. E. S.; BUILES, J. A. J. Madce - tvd - model agile development educational content for digital television. **IEEE Latin America Transactions**, v. 13, n. 10, p. 3432–3438, Oct 2015. ISSN 1548-0992. Citado na página 49.

GIPPLE, J.; LORD, E. **Understanding mobile learning and best practices**. ICS Learning Group. 2016. Citado na página 29.

GOOGLE. Techreport, **Introducing material design**. <https://developer.android.com/design/>: [s.n.], 2018. Developer Android. Available at <https://developer.android.com/design/>. Accessed at 05/04/2018. Disponível em: <<https://developer.android.com/design/>>. Citado nas páginas 59, 60, 61, 62, 64, 65 e 66.

HASHEMI, M.; AZIZINEZHAD, M.; NAJAFI, V.; NESARI, A. J. What is mobile learning? challenges and capabilities. **Procedia-Social and Behavioral Sciences**, Elsevier, v. 30, p. 2477–2481, 2011. Citado na página 35.

HEFLIN, H.; SHEWMAKER, J.; NGUYEN, J. Impact of mobile technology on student attitudes, engagement, and learning. **Computers & Education**, Elsevier, v. 107, p. 91–99, 2017. Citado nas páginas 29, 34 e 35.

HOWARD, S. K.; TONDEUR, J.; MA, J.; YANG, J. What to teach? strategies for developing digital competency in preservice teacher training. **Computers & Education**, Elsevier, p. 104149, 2021. Citado na página 29.

HRYCYK, J. C.; SOARES, I. W.; AGNER, L. T. W. Definition of object constraint language rules in models to support the development of android applications. **J. Comput. Sci.**, v. 14, n. 2, p. 253–259, 2018. Citado na página 30.

HWANG, G.-J.; WU, P.-H.; KE, H.-R. An interactive concept map approach to supporting mobile learning activities for natural science courses. **Computers & Education**, Elsevier, v. 57, n. 4, p. 2272–2280, 2011. Citado na página 35.

IBGE. **Pesquisa Nacional por Amostra de Domicílios - 2018: síntese de indicadores**. [S.l.], 2018. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101705_informativo.pdf>. Citado na página 28.

INTELLIGENCE, G. **2020 Mobile Industry Impact Report: Sustainable Development Goals**. 2020. Citado na página 28.

IQBAL, S.; BHATTI, Z. A qualitative exploration of teachers' perspective on smartphones usage in higher education in developing countries. **International Journal of Educational Technology in Higher Education**, v. 17, n. 1, 2020. Cited By 0. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85087414399&doi=10.1186%2fs41239-020-00203-4&partnerID=40&md5=0dcc3d1088b950e9ee5616eae3d975b2>>. Citado na página 35.

IVANOV, I.; BÉZIVIN, J.; AKSIT, M. Technological spaces: An initial appraisal. In: **4th International Symposium on Distributed Objects and Applications, DOA 2002**. [S.l.: s.n.], 2002. Citado nas páginas 30 e 56.

JAHNKE, I.; LIEBSCHER, J. Three types of integrated course designs for using mobile technologies to support creativity in higher education. **Computers & Education**, Elsevier, v. 146, p. 103782, 2020. Citado nas páginas 27, 29 e 121.

JUGEN, Y.; RUONAN, X. Mobile terminal based mobile learning system design. In: **IEEE. Computer Science & Education (ICCSE), 2016 11th International Conference on**. [S.l.], 2016. p. 699–703. Citado na página 33.

- JÚNIOR, C. G. F.; SOARES, J. M.; BARROSO, G. C.; CASTRO, L.; SOARES, M. I. d. S.; JOYE, C. R.; FREITAS, A. T. d.; SOARES, É. F. Um modelo para a produção de Objetos de Aprendizagem Acessíveis: Modelagem e Análise por Redes de Petri Coloridas. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 25, n. 1, p. 1143, 2014. ISSN 2316-6533. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/3060>>. Citado nas páginas 28 e 49.
- KEMCZINSKI, A.; COSTA, I. A.; WEHRMEISTER, M. A.; HOUNSELL, M. d. S.; VAHL-DICK, A. Metodologia para Construção de Objetos de Aprendizagem Interativos. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 23, n. 1, 2012. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1739>>. Citado nas páginas 28 e 49.
- KENSKI, V. M. **Educação e tecnologias**. [S.l.]: Papyrus editora, 2007. Citado na página 27.
- KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007. Citado na página 46.
- KLEPPE, A.; WARMER, J.; BAST, W. **MDA Explained: The Model Driven Architecture: Practice and Promise**. 1st. ed. [S.l.]: Addison-Wesley, 2003. 170 p. Citado nas páginas 15, 43 e 45.
- KLEPSCH, M.; SEUFERT, T. Understanding instructional design effects by differentiated measurement of intrinsic, extraneous, and germane cognitive load. **Instructional Science**, Springer, v. 48, n. 1, p. 45–77, 2020. Citado nas páginas 59, 60 e 66.
- LABIB, A. E.; PENADÉS, M. C.; CANÓS, J. H.; GÓMEZ, A. Enforcing reuse and customization in the development of learning objects: A product line approach. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2015. (SAC '15), p. 261–263. ISBN 978-1-4503-3196-8. Disponível em: <<http://doi.acm.org/10.1145/2695664.2695991>>. Citado nas páginas 49 e 52.
- Lafi, L.; Feki, J.; Hammoudi, S. Metamodel matching techniques evaluation and benchmarking. In: **2013 International Conference on Computer Applications Technology (ICCAT)**. [S.l.: s.n.], 2013. p. 1–6. Citado na página 42.
- LANG, J.; ČERVENÁK, M. Use case driven educational content modeling with uml. In: **2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)**. [S.l.: s.n.], 2014. p. 289–294. Citado na página 49.
- LAPOLL, F. R.; MOTTA, C. L. R.; CRUZ, C. M.; OLIVEIRA, C. E. T. Modelo de Desenvolvimento de Objetos de Aprendizagem Baseado em Metodologias Ágeis. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 1, n. 1, nov. 2009. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1197>>. Citado na página 49.
- Latif, M.; Lakhrissi, Y.; Nfaoui, E. H.; Es-Sbai, N. Review of mobile cross platform and research orientations. In: **2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)**. [S.l.: s.n.], 2017. p. 1–4. Citado nas páginas 30 e 77.
- LOPES, D.; HAMMOUDI, S.; ABDELOUAHAB, Z. Schema Matching in the Context of Model Driven Engineering: From Theory to Practice. **Proceedings of the International Conference on Systems, Computing Sciences and Software Engineering (SCSS 2005)**, December 2005. Citado na página 42.

- LOPES, D. C. P. **Introdução a Engenharia Dirigida por Modelos**. [S.l.], 2007. Citado na página 43.
- LOUHAB, F. E.; BAHNASSE, A.; TALEA, M. Considering mobile device constraints and context-awareness in adaptive mobile learning for flipped classroom. **Education and Information Technologies**, Springer, v. 23, n. 6, p. 2607–2632, 2018. Citado na página 28.
- MACHADO, J. B.; ISOTANI, S.; BARBOSA, E. F.; RIBEIRO, M. B. Towards an Ontological Infrastructure for Content Modeling and Personalization. In: **2012 Seventh International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)**. [S.l.: s.n.], 2012. p. 107–112. Citado na página 30.
- MADARIAGA, L.; NUSSBAUM, M.; GUTIÉRREZ, I.; BARAHONA, C.; MENESES, A. Assessment of user experience in video-based learning environments: From design guidelines to final product. **Computers & Education**, Elsevier, p. 104176, 2021. Citado nas páginas 65 e 66.
- MARCOLINO, A. S.; BARBOSA, E. Towards an m-learning requirements catalog for the development of educational applications for the teaching of programming. In: **IEEE. Frontiers in Education Conference (FIE), 2016 IEEE**. [S.l.], 2016. p. 1–5. Citado nas páginas 27 e 33.
- MARTIN, F.; ERTZBERGER, J. Here and now mobile learning: An experimental study on the use of mobile technology. **Computers & Education**, Elsevier, v. 68, p. 76–85, 2013. Citado na página 34.
- MARTINEZ, D.; FERRE, X.; GUERRERO, G.; JURISTO, N. An agile-based integrated framework for mobile application development considering ilities. **IEEE Access**, IEEE, v. 8, p. 72461–72470, 2020. Citado na página 30.
- MAYER, R. Using multimedia for e-learning. **Journal of Computer Assisted Learning**, v. 33, n. 5, p. 403–423, 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12197>>. Citado nas páginas 58, 65 e 66.
- MAYER, R. E.; FIORELLA, L.; STULL, A. Five ways to increase the effectiveness of instructional video. **Educational Technology Research and Development**, Springer, p. 1–16, 2020. Citado nas páginas 27, 59, 62, 65 e 66.
- MERRILL, M. D. Component display theory. **Instructional-design theories and models: An overview of their current status**, v. 1, p. 282–333, 1983. Citado nas páginas 39 e 71.
- MICHAELIS. **Brazilian Portuguese Language Dictionary**. 2020. Last accessed 16 April 2020. Disponível em: <<http://michaelis.uol.com.br>>. Citado na página 39.
- MINOVIĆ, M.; MILOVANOVIĆ, M.; STARCEVIC, D.; JOVANOVIĆ, M. Knowledge modeling for educational games. In: SPRINGER. **World Summit on Knowledge Society**. [S.l.], 2009. p. 156–165. Citado nas páginas 49, 51 e 52.
- MORAES, M.; RAYMUNDO, V.; BOCCHESI, J.; LIMA, V. Elaboração de objetos de aprendizagem para o lapren: Processo de desenvolvimento e sistema de produção. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2011. v. 1, n. 1. Citado nas páginas 49 e 52.
- MUTLU-BAYRAKTAR, D.; COSGUN, V.; ALTAN, T. Cognitive load in multimedia learning environments: A systematic review. **Computers & Education**, Elsevier, v. 141, p. 103618, 2019. Citado na página 59.

- MYLLYMÄKI, M.; HAKALA, I.; HÄRMÄNMAA, T. A new way to produce video-based learning material. In: IEEE. **2018 28th EAEEIE Annual Conference (EAEEIE)**. [S.l.], 2018. p. 1–9. Citado nas páginas 65 e 66.
- NIELSEN, J.; BUDIU, R. **Mobile usability**. [S.l.]: MITP-Verlags GmbH & Co. KG, 2013. Citado nas páginas 60 e 66.
- NOGUEIRA, U. S.; AMORIM, R. J. R.; LÓSCIO, B. F. Desenvolvimento de uma ontologia para recursos educacionais abertos e fragmentados usando a abordagem methontology. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2014. v. 25, n. 1, p. 402. Citado na página 49.
- NOVAK, J. D. Concept mapping: A useful tool for science education. **Journal of research in science teaching**, Wiley Online Library, v. 27, n. 10, p. 937–949, 1990. Citado na página 38.
- OLIVEIRA, E. R.; NELSON, M. A. V.; ISHITANI, L. Ciclo de vida de objetos de aprendizagem baseado no padrão scorm. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2007. v. 1, n. 1, p. 217–226. Citado na página 49.
- OLIVEIRA, J.; LOPES, D.; ABDELOUAHAB, Z.; CLARO, D.; HAMMOUDI, S. Model driven testing for cloud computing. In: **Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering**. [S.l.]: Springer, 2015. p. 297–304. Citado na página 30.
- OMG. **UML Version 2.3 Specification, OMG documents formal/2010-05-03**. <http://www.omg.org/spec/UML/2.3/>: [s.n.], 2010. Citado na página 44.
- OZDAMLI, F.; CAVUS, N. Basic elements and characteristics of mobile learning. **Procedia-Social and Behavioral Sciences**, Elsevier, v. 28, p. 937–942, 2011. Citado na página 35.
- PAULINS, N.; BALINA, S.; ARHIPOVA, I. Learning content development methodology for mobile devices. **Procedia Computer Science**, v. 43, p. 147 – 153, 2015. ISSN 1877-0509. ICTE in Regional Development, December 2014, Valmiera, Latvia. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050914015889>>. Citado nas páginas 37 e 59.
- PESSOA, M. de C.; BENITTI, F. B. V. Proposta de um processo para produção de objetos de aprendizagem. **HÍFEN**, v. 32, n. 62, 2008. Citado na página 49.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 55.
- PRIETO, J. C. S.; MIGUELÁÑEZ, S. O.; GARCÍA-PEÑALVO, F. J. Mobile learning: tendencies and lines of research. In: ACM. **Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality**. [S.l.], 2013. p. 473–480. Citado na página 28.
- REIS, E. A.; REIS, I. A. Análise descritiva de dados. **Relatório Técnico do Departamento de Estatística da UFMG**, v. 1, 2002. Citado na página 106.
- ROCCO, J. D.; RUSCIO, D. D.; IOVINO, L.; PIERANTONIO, A. Collaborative repositories in model-driven engineering [software technology]. **IEEE Software**, v. 32, n. 3, p. 28–34, May 2015. ISSN 0740-7459. Citado na página 30.

RODRÍGUEZ, A. I.; RIAZA, B. G.; GÓMEZ, M. C. S. Collaborative learning and mobile devices: An educational experience in primary education. **Computers in Human Behavior**, Elsevier, v. 72, p. 664–677, 2017. Citado na página 27.

RUBACH, C.; LAZARIDES, R. Addressing 21st-century digital skills in schools—development and validation of an instrument to measure teachers’ basic ict competence beliefs. **Computers in Human Behavior**, Elsevier, v. 118, p. 106636, 2021. Citado na página 29.

SANTOS, N. d. S. R. S. d.; WIVES, L.; LIMA, J. V. D.; SILVA, A. H. d. Avaliação do modelo m-ROAMI: um experimento para a análise do seu potencial de ensino, interatividade e ubiquidade. v. 27, n. 1, p. 220, 2016. ISSN 2316-6533. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/6702>>. Citado na página 33.

SARRAB, M.; AL-SHIHI, H.; AL-MANTHARI, B.; BOURDOUCEN, H. Toward educational requirements model for mobile learning development and adoption in higher education. **TechTrends**, v. 62, n. 6, p. 635–646, 2018. Disponível em: <<https://doi.org/10.1007/s11528-018-0331-4>>. Citado nas páginas 33 e 121.

SILVA, M. A. G.; BARBOSA, E. F.; MALDONADO, J. C. Model-driven development of learning objects. In: **2011 Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2011. p. F4E–1–F4E–6. Citado nas páginas 30, 49, 51 e 52.

SOMMERVILLE, I. Engenharia de software, 9ª edição, tradução: Selma shin shimizu mel-nikoff, reginaldo arakaki, edilson de andrade barbosa. **São Paulo: Pearson Addison-Wesley**, v. 22, p. 103, 2011. Citado na página 42.

SOUZA, M. d. F. C. d.; FILHO, J. A. d. C.; ANDRADE, R. M. C. Customização Guiada: uma estratégia Orientada a Modelos para a Produção de Objetos de Aprendizagem. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 23, n. 1, 2012. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1807>>. Citado nas páginas 49, 51 e 52.

SRIVASTAVA, S.; GULATI, V. P. M-learning—on path to integration with organisation systems. **International Association for Development of the Information Society**, ERIC, 2014. Citado na página 28.

STEINBERG, D.; BUDINSKY, F.; MERKS, E.; PATERNOSTRO, M. **EMF: eclipse modeling framework**. [S.l.]: Pearson Education, 2008. Citado nas páginas 46 e 86.

SÁNCHEZ, R. A.; CORTIJO, V.; JAVED, U. Students’ perceptions of facebook for academic purposes. **Computers & Education**, v. 70, p. 138 – 149, 2014. ISSN 0360-1315. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360131513002340>>. Citado na página 28.

TONDEUR, J.; HOWARD, S. K.; YANG, J. One-size does not fit all: Towards an adaptive model to develop preservice teachers’ digital competencies. **Computers in Human Behavior**, Elsevier, v. 116, p. 106659, 2021. Citado na página 29.

UNESCO, O. d. N. U. p. a. E. **UNESCO Policy Guidelines for Mobile Learning**. [S.l.], 2014. Disponível em: <<http://unesdoc.unesco.org/images/0022/002277/227770por.pdf>>. Citado na página 34.

UNION, I. I. T. **International Telecommunication Union - ITU**. 2020. Citado na página 121.

WANG, M.; SHEN, R. Message design for mobile learning: Learning theories, human cognition and design principles. **British Journal of Educational Technology**, Wiley Online Library, v. 43, n. 4, p. 561–575, 2012. Citado na página 59.

WANG, Y.; LU, H. Validating items of different modalities to assess the educational technology competency of pre-service teachers. **Computers & Education**, Elsevier, v. 162, p. 104081, 2021. Citado na página 29.

WASTIAU, P.; BLAMIRE, R.; KEARNEY, C.; QUITTRE, V.; GAER, E. Van de; MONSEUR, C. The use of ict in education: a survey of schools in europe. **European Journal of Education**, Wiley Online Library, v. 48, n. 1, p. 11–27, 2013. Citado na página 27.

WHITTLE, J.; HUTCHINSON, J.; ROUNCEFIELD, M. The state of practice in model-driven engineering. **IEEE Software**, v. 31, n. 3, p. 79–85, May 2014. ISSN 0740-7459. Citado na página 30.

IMPLEMENTAÇÃO DE REGRAS DE TRANSFORMAÇÃO DE MODELOS

A.1 Instância do Modelo Educom

Um Modelo Educom é instanciado a partir da base de dados da ferramenta **Educom Tool**. A regra de transformação a seguir busca as informações do modelada pelo professor na **Educom Tool** gerando um Modelo Educom de um domínio específico.

Código-fonte 2 – Regras de transformação em Java - Modelo Educom

```
1 package br.com.educom.dao;
2 import java.io.IOException;
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.util.ArrayList;
8 import java.util.Collection;
9 import java.util.List;
10 import org.eclipse.emf.common.util.BasicEList;
11 import org.eclipse.emf.common.util.EList;
12 import org.eclipse.emf.common.util.URI;
13 import org.eclipse.emf.ecore.resource.Resource;
14 import org.eclipse.emf.ecore.resource.ResourceSet;
15 import org.eclipse.emf.ecore.resource.impl.ResourceSetImpl;
16 import org.eclipse.emf.ecore.xmi.impl.XMIResourceFactoryImpl;
17 import mmInstructional.AnswersAlternatives;
18 import mmInstructional.Audio;
19 import mmInstructional.CollaborativeActivity;
20 import mmInstructional.Concept;
21 import mmInstructional.Fact;
22 import mmInstructional.IConcept;
23 import mmInstructional.IKnowledgeDomain;
24 import mmInstructional.Image;
25 import mmInstructional.InformationItem;
26 import mmInstructional.InstrucElementType;
```

```

27 import mmInstructional.InstructionalElement;
28 import mmInstructional.Link;
29 import mmInstructional.MmInstructionalFactory;
30 import mmInstructional.MmInstructionalPackage;
31 import mmInstructional.MobileMedia;
32 import mmInstructional.Module;
33 import mmInstructional.Phase;
34 import mmInstructional.Principle;
35 import mmInstructional.Procedure;
36 import mmInstructional.Question;
37 import mmInstructional.Text;
38 import mmInstructional.Video;
39
40 public class GetDomain {
41     static MmInstructionalFactory instructionalFactory = MmInstructionalFactory.
42         eINSTANCE;
43     static MmInstructionalPackage instructionalPackage = MmInstructionalPackage.
44         eINSTANCE;
45     private Connection conn;
46     public GetDomain(Connection conn) {
47         super();
48         this.conn = conn;
49     }
50     private PreparedStatement stmt;
51     private BasicEList<Module> listModules;
52     private BasicEList<IConcept> listConcepts;
53     private IKnowledgeDomain domain = instructionalFactory.createIKnowledgeDomain
54         ();
55     public IKnowledgeDomain getDomain(int idKDomain) throws SQLException {
56         String author;
57         stmt = this.conn.prepareStatement(
58             "SELECT * FROM public.\"KnowledgeDomain\" where \"idKnowledgeDomain\" =
59             " + idKDomain);
60         ResultSet rs = stmt.executeQuery();
61         boolean rsIsEmpty = false;
62         IKnowledgeDomain domain = instructionalFactory.createIKnowledgeDomain();
63         while (rs.next()) {
64             domain = instructionalFactory.createIKnowledgeDomain();
65             domain.setNameIKDomain(rs.getString("nameKnowledgeDomain"));
66             domain.setSubTitle(rs.getString("subTitle"));
67             int idUser = rs.getInt("fk_idUser");
68             author = getAuthor(idUser);
69             domain.setAuthor(author);
70             domain.getHasIElementType().addAll(getListInstrucElementTypes());
71             setDomain(domain);
72             domain.getIHasModule().addAll(getListModule(rs.getInt("idKnowledgeDomain")
73             ));
74             setDomain(domain);
75             rsIsEmpty = true;
76         }
77         return domain;
78     }
79     private String getAuthor(int idUser) {
80         String first_name = "";
81         String last_name = "";

```

```

79     try {
80         stmt = this.conn
81             .prepareStatement("SELECT first_name, last_name FROM public.core_user
where id = " + idUser);
82         ResultSet rs = stmt.executeQuery();
83         while (rs.next()) {
84             first_name = rs.getString("first_name");
85             last_name = rs.getString("last_name");
86             first_name = first_name.substring(0, 1).toUpperCase() + first_name.
substring(1);
87             last_name = last_name.substring(0, 1).toUpperCase() + last_name.
substring(1);
88         }
89     } catch (SQLException e) {
90         e.printStackTrace();
91     }
92     return first_name + " " + last_name;
93 }
94
95 public EList<IKnowledgeDomain> getListIKDomain() {
96     BasicEList<IKnowledgeDomain> ikDomains = new BasicEList<IKnowledgeDomain>();
97     try {
98         stmt = this.conn.prepareStatement("SELECT * FROM public.\"KnowledgeDomain
\"");
99         ResultSet rs = stmt.executeQuery();
100        while (rs.next()) {
101            IKnowledgeDomain ikd = instructionalFactory.createIKnowledgeDomain();
102            ikd.setNameIKDomain(rs.getString("nameKnowledgeDomain"));
103            ikd.setSubTitle(rs.getString("subTitle"));
104            ikd.getHasIElementType().addAll(getListInstrucElementTypes());
105            setDomain(ikd);
106            ikd.getIHasModule().addAll(getListModule(rs.getInt("idKnowledgeDomain"))
);
107            setDomain(ikd);
108            ikDomains.add(ikd);
109        }
110    } catch (SQLException e) {
111        throw new RuntimeException(e);
112    }
113    return ikDomains;
114 }
115
116 public List<InstrucElementType> getListInstrucElementTypes() {
117     List<InstrucElementType> listIet = new ArrayList<InstrucElementType>();
118     try {
119         stmt = this.conn.prepareStatement("SELECT * FROM public.\"
InstrucElementType\"");
120         ResultSet rs = stmt.executeQuery();
121         while (rs.next()) {
122             InstrucElementType iet = instructionalFactory.createInstrucElementType()
;
123             iet.setName(rs.getString("nameInstrucElementType"));
124             listIet.add(iet);
125         }
126     } catch (SQLException e) {
127         e.printStackTrace();
128     }

```

```

129     return listIet;
130 }
131
132 public EList<Module> getListModule(int idKnowledgeDomain) {
133     BasicEList<Module> modules = new BasicEList<Module>();
134     Module moduleSuper = instructionalFactory.createModule();
135     int index = -1;
136     try {
137         stmt = this.conn.prepareStatement("SELECT * FROM public.\"Module\" where
138     \"idKnowledgeDomain\" = "
139         + idKnowledgeDomain + "ORDER BY \"idModule\" asc , \"fk_IdModule\" asc
140     ");
141         ResultSet rs = stmt.executeQuery();
142         while (rs.next()) {
143             Module obj = instructionalFactory.createModule();
144             obj.setLabel(rs.getString("nameModule"));
145             obj.setNameModule(rs.getString("nameModule"));
146             obj.setVisible(rs.getBoolean("visible"));
147             System.out.println("Module: " + rs.getInt("idModule") + " " + rs.
148             getString("nameModule"));
149             Collection<? extends MobileMedia> listMobileMedia = getMobileMediaModule
150             (rs.getInt("idModule"));
151             obj.getModuleToMMedia().addAll(listMobileMedia);
152             obj.getIModuleHasConcepts()
153             .addAll(getListIConcept(rs.getInt("idKnowledgeDomain"), rs.getInt("
154             idModule")));
155             obj.getModuleToIElement().addAll(getInstructionaElementModule(rs.getInt(
156             "idModule")));
157             // É um submodulo?
158             if (rs.getInt("fk_IdModule") > 0) {
159                 stmt = this.conn.prepareStatement("SELECT * FROM public.\"Module\"
160                 where \"idModule\" = "
161                 + rs.getInt("fk_IdModule") + "ORDER BY \"idModule\" asc");
162                 ResultSet rstmp = stmt.executeQuery();
163                 while (rstmp.next()) {
164                     for (Module module : modules) {
165                         if (module.getNameModule().equals(rstmp.getString("nameModule")))
166                         {
167                             obj.getModuleToMMedia().addAll(getMobileMediaModule(rstmp.getInt
168                             ("idModule")));
169                             module.getSubModule().add(obj);
170                             index = modules.indexOf(module);
171                             moduleSuper = module;
172                         }
173                     }
174                 }
175                 if (index != -1) {
176                     modules.remove(index);
177                     modules.add(index, moduleSuper);
178                 }
179             }
180             } else {
181                 modules.add(obj);
182             }
183         }
184     } catch (SQLException e) {
185         e.printStackTrace();
186     }

```

```

177     setListModules(modules);
178     return modules;
179 }
180
181 private ResultSet getResultSetAll(String pColuns, String pTable, String pWhere
    , int value, String pOrderBy)
182     throws SQLException {
183     String coluns = pColuns;
184     String table = pTable;
185     String where = pWhere;
186     String orderBy = pOrderBy;
187     String query = "SELECT" + coluns + " " + "FROM public." + table + "where "
    + where + "=" + value + "ORDER BY "
188         + orderBy + " asc";
189     stmt = this.conn.prepareStatement(query);
190     ResultSet rs = stmt.executeQuery();
191     return rs;
192 }
193
194 public EList<IConcept> getListIConcept(int idDomain, int idModule) throws
    SQLException {
195     String q = "SELECT * FROM public.\"Concept\" where \"fk_idKnowledgeDomain\"
    =" + idDomain
196         + "and \"fk_idModule\" = " + idModule + "ORDER BY \"idConcept\"";
197     stmt = this.conn.prepareStatement(q);
198     ResultSet rs = stmt.executeQuery();
199     BasicEList<IConcept> iconcepts = new BasicEList<IConcept>();
200     while (rs.next()) {
201         IConcept iconcept = instructionalFactory.createIConcept();
202         iconcept.setNameIConcept(rs.getString("nameConcept"));
203         iconcept.setVisible(rs.getBoolean("visible"));
204         iconcept.getHasInforItem()
205             .addAll(getListInfomationItem(rs.getInt("idConcept"), rs.getString("
    nameConcept")));
206         iconcept.getHasInstrucinfors().addAll(getInstructionaElementConcept(rs.
    getInt("idConcept")));
207         iconcepts.add(iconcept);
208     }
209     setListConcepts(iconcepts);
210     return iconcepts;
211 }
212
213 private Collection<? extends InstructionalElement>
    getInstructionaElementConcept(int idConcept)
214     throws SQLException {
215     String q = "SELECT * FROM public.\"InstructionalElement\" where \"
    fk_idConcept\" = " + idConcept;
216     stmt = this.conn.prepareStatement(q);
217     ResultSet rs = stmt.executeQuery();
218     return getListInstrucionalElement(rs);
219 }
220
221 private Collection<? extends InstructionalElement>
    getInstructionaElementModule(int idModule) throws SQLException {
222     String q = "SELECT * FROM public.\"InstructionalElement\" where \"
    fk_idModule\" = " + idModule;
223     stmt = this.conn.prepareStatement(q);

```

```

224     ResultSet rs = stmt.executeQuery();
225     return (BasicEList<InstructionalElement>) getListInstrucionalElement(rs);
226 }
227
228 private Collection<? extends InstructionalElement> getListInstrucionalElement
229 (ResultSet rs) throws SQLException {
230     BasicEList<InstructionalElement> listInstElem = new BasicEList<
231     InstructionalElement>();
232     while (rs.next()) {
233         InstructionalElement instElement = instructionalFactory.
234         createInstructionalElement();
235         InstrucElementType elementType = instructionalFactory.
236         createInstrucElementType();
237         instElement.setLabel(rs.getString("label"));
238         instElement.setVisible(rs.getBoolean("visible"));
239         switch (rs.getInt("fk_InstrucionalElementType")) {
240             case 1: // exercicio
241                 elementType = domain.getHasIElementType().get(0);
242                 instElement.setHasIElementType(elementType);
243                 instElement.getIElementQuestion().addAll(getListQuestion(rs.getInt("
244                 idInstrucionalElement")));
245                 listInstElem.add(instElement);
246                 break;
247             case 2: // avaliacao
248                 elementType = domain.getHasIElementType().get(1);
249                 instElement.setHasIElementType(elementType);
250                 instElement.getIElementQuestion().addAll(getListQuestion(rs.getInt("
251                 idInstrucionalElement")));
252                 listInstElem.add(instElement);
253                 break;
254             case 3: // atividade colaborativa
255                 elementType = domain.getHasIElementType().get(2);
256                 instElement.setHasIElementType(elementType);
257                 instElement.getIElementCollab().addAll(getListCollaborative(rs.getInt("
258                 idInstrucionalElement")));
259                 listInstElem.add(instElement);
260                 break;
261             default:
262                 break;
263         }
264     }
265     return listInstElem;
266 }
267
268 private Collection<? extends CollaborativeActivity> getListCollaborative(int
269 idInstrucionalElement)
270 throws SQLException {
271     String q = "SELECT * FROM public.\"InstructionalElement\" where \"
272     idInstrucionalElement\" = "

```



```

271     + idInstructionalElement;
272     stmt = this.conn.prepareStatement(q);
273     ResultSet rs = stmt.executeQuery();
274     BasicEList<CollaborativeActivity> collaborativeActivities = new BasicEList<
CollaborativeActivity>();
275     while (rs.next()) {
276         CollaborativeActivity activity = instructionalFactory.
createCollaborativeActivity();
277         activity.setLabel(rs.getString("label"));
278         activity.setMemberAmount(rs.getInt("memberAmount"));
279         Text text = instructionalFactory.createText();
280         text.setTextFull(rs.getString("description"));
281         text.setLabel(rs.getString("label"));
282         text.setVisible(rs.getBoolean("visible"));
283         activity.getColActivityToMMedia().add(text);
284         activity.getColActivityToMMedia().addAll(getMobileMediaInstrucionalElement
(idInstructionalElement));
285         activity.getCollaborativeToQuestion().addAll(getListQuestion(rs.getInt("
idInstructionalElement"))));
286         collaborativeActivities.add(activity);
287     }
288     return collaborativeActivities;
289 }
290
291 private Collection<? extends Question> getListQuestion(int
idInstructionalElement) throws SQLException {
292     String q = "SELECT * FROM public.\"Question\" where \"
fk_idInstructionalElement\" = " + idInstructionalElement;
293     stmt = this.conn.prepareStatement(q);
294     ResultSet rs = stmt.executeQuery();
295     BasicEList<Question> questions = new BasicEList<Question>();
296     while (rs.next()) {
297         Question question = instructionalFactory.createQuestion();
298         question.setDescriptionQuestion(rs.getString("descriptionQuestion"));
299         question.setOrderQuestion(rs.getInt("orderQuestion"));
300         question.setTypeQuestion(rs.getInt("typeQuestion"));
301         question.getQuestionToAnswers().addAll(getListAnswers(rs.getInt("
idQuestion"))));
302         questions.add(question);
303     }
304     return questions;
305 }
306
307 private Collection<? extends AnswersAlternatives> getListAnswers(int
idQuestion) throws SQLException {
308     String q = "SELECT * FROM public.\"AnswersAlternatives\" where \"
fk_idQuestion\" = " + idQuestion
309         + "ORDER BY \"fk_idQuestion\" asc, \"idObjAnswer\" asc";
310     stmt = this.conn.prepareStatement(q);
311     ResultSet rs = stmt.executeQuery();
312     BasicEList<AnswersAlternatives> alternatives = new BasicEList<
AnswersAlternatives>();
313     while (rs.next()) {
314         AnswersAlternatives answersAlternatives = instructionalFactory.
createAnswersAlternatives();
315         answersAlternatives.setIdObjAnswer(rs.getString("idObjAnswer"));
316         answersAlternatives.setAnswers(rs.getString("answers"));

```

```

317     answersAlternatives.setIsTrue(rs.getBoolean("isTrue"));
318     alternatives.add(answersAlternatives);
319 }
320 return alternatives;
321 }
322
323 private Collection<? extends InformationItem> getListInformationItem(int
    idConcept, String nameIConcept)
324     throws SQLException {
325     String q = "SELECT * FROM public.\"InformationItem\" where \"fk_idConcept\"
    = " + idConcept
326         + "ORDER BY \"idInformationItem\" asc";
327     stmt = this.conn.prepareStatement(q);
328     ResultSet rs = stmt.executeQuery();
329     BasicEList<InformationItem> listInfItem = new BasicEList<InformationItem>();
330     while (rs.next()) {
331         int informationType = rs.getInt("fk_InformationItemType");
332         int idInformationItem = rs.getInt("idInformationItem");
333         switch (informationType) {
334             case 1: // concept
335                 Concept concept = instructionalFactory.createConcept();
336                 concept.getConceptHasMMedia().addAll(getMobileMediaConcept(
    idInformationItem));
337                 concept.setLabel(nameIConcept);
338                 listInfItem.add(concept);
339                 break;
340             case 2: // principle
341                 Principle princple = instructionalFactory.createPrinciple();
342                 princple.getPrincToMMedia().addAll(getMobileMediaConcept(
    idInformationItem));
343                 princple.setLabel(nameIConcept);
344                 listInfItem.add(princple);
345                 break;
346             case 3: // fact
347                 Fact fact = instructionalFactory.createFact();
348                 fact.getFactToMMedia().addAll(getMobileMediaConcept(idInformationItem));
349                 fact.setLabel(nameIConcept);
350                 listInfItem.add(fact);
351                 break;
352             case 4: // procedure
353                 Procedure procedure = instructionalFactory.createProcedure();
354                 procedure.getProcToMMedia().addAll(getMobileMediaConcept(
    idInformationItem));
355                 procedure.setLabel(nameIConcept);
356                 procedure.getPhase().addAll(getListPhase(rs.getInt("idInformationItem"))
    );
357                 listInfItem.add(procedure);
358                 break;
359             default:
360                 break;
361         }
362     }
363     return listInfItem;
364 }
365
366 private Collection<? extends Phase> getListPhase(int fk_InformationItem)
    throws SQLException {

```

```
367     String q = "SELECT * FROM public.\"PhaseProcedure\" where \"
368     fk_InformationItem\" =\" + fk_InformationItem;
369     stmt = this.conn.prepareStatement(q);
370     ResultSet rs = stmt.executeQuery();
371     BasicEList<Phase> phases = new BasicEList<Phase>();
372     while (rs.next()) {
373         Phase phase = instructionalFactory.createPhase();
374         phase.setDescription(rs.getString("description"));
375         phase.setOrdem(rs.getInt("order"));
376         phases.add(phase);
377     }
378     return phases;
379 }
380 private Collection<? extends MobileMedia> getMobileMediaInstrucionalElement(
381     int idInstrucionalElement)
382     throws SQLException {
383     String q = "SELECT * FROM public.\"MobileMedia\" where \"
384     fk_idInstrucionalElement\" =\" + idInstrucionalElement
385     + "ORDER BY \"fk_idInstrucionalElement\" asc";
386     stmt = this.conn.prepareStatement(q);
387     ResultSet rs = stmt.executeQuery();
388     BasicEList<MobileMedia> mobilemedias = new BasicEList<MobileMedia>();
389     mobilemedias.addAll(getMobileMedia(rs));
390     return mobilemedias;
391 }
392 private Collection<? extends MobileMedia> getMobileMediaConcept(int
393     idInformationItem) throws SQLException {
394     String q = "SELECT * FROM public.\"MobileMedia\" where \"fk_Informationitem
395     \" =\" + idInformationItem
396     + "ORDER BY \"fk_Informationitem\" asc";
397     stmt = this.conn.prepareStatement(q);
398     ResultSet rs = stmt.executeQuery();
399     BasicEList<MobileMedia> mobilemedias = new BasicEList<MobileMedia>();
400     mobilemedias.addAll(getMobileMedia(rs));
401     return mobilemedias;
402 }
403 private Collection<? extends MobileMedia> getMobileMediaModule(int idModulo)
404     throws SQLException {
405     String q = "SELECT * FROM public.\"MobileMedia\" where \"fk_Module\"=\" +
406     idModulo
407     + "ORDER BY \"idMobileMedia\" asc";
408     stmt = this.conn.prepareStatement(q);
409     ResultSet rs = stmt.executeQuery();
410     BasicEList<MobileMedia> mobilemedias = new BasicEList<MobileMedia>();
411     mobilemedias.addAll(getMobileMedia(rs));
412     return mobilemedias;
413 }
414 private Collection<? extends MobileMedia> getMobileMedia(ResultSet rs) throws
415     SQLException {
416     BasicEList<MobileMedia> listMobileMedia = new BasicEList<MobileMedia>();
417     while (rs.next()) {
418         switch (rs.getInt("fk_idMediaType")) {
419             case 1: // image
```

```
416 Image image = instructionalFactory.createImage();
417 image.setLabel(rs.getString("description"));
418 image.setDescription(rs.getString("description"));
419 image.setNameFile(rs.getString("nameFile"));
420 image.setPath(rs.getString("path"));
421 image.setDifficultyLevel(rs.getInt("difficultyLevel"));
422 image.setLearningStyle(rs.getInt("learningStyle"));
423 image.setResolution(rs.getString("resolution"));
424 image.setVisible(rs.getBoolean("visible"));
425 listMobileMedia.add(image);
426 break;
427 case 2: // video
428 Video video = instructionalFactory.createVideo();
429 video.setLabel(rs.getString("description"));
430 video.setNameFile(rs.getString("nameFile"));
431 video.setDifficultyLevel(rs.getInt("difficultyLevel"));
432 video.setLearningStyle(rs.getInt("learningStyle"));
433 video.setResolution(rs.getString("resolution"));
434 video.setVisible(rs.getBoolean("visible"));
435 video.setPath(rs.getString("path"));
436 video.setResolution(rs.getString("resolution"));
437 video.setTime(Double.parseDouble(rs.getString("time")));
438 listMobileMedia.add(video);
439 break;
440 case 3: // audio
441 Audio audio = instructionalFactory.createAudio();
442 audio.setLabel(rs.getString("description"));
443 audio.setNameFile(rs.getString("nameFile"));
444 audio.setDifficultyLevel(rs.getInt("difficultyLevel"));
445 audio.setLearningStyle(rs.getInt("learningStyle"));
446 audio.setVisible(rs.getBoolean("visible"));
447 audio.setPath(rs.getString("path"));
448 listMobileMedia.add(audio);
449 break;
450 case 4: // text
451 Text text = instructionalFactory.createText();
452 text.setLabel(rs.getString("label"));
453 text.setTextFull(rs.getString("textFull"));
454 text.setTextShort(rs.getString("textShort"));
455 text.setDifficultyLevel(rs.getInt("difficultyLevel"));
456 text.setLearningStyle(rs.getInt("learningStyle"));
457 text.setVisible(rs.getBoolean("visible"));
458 listMobileMedia.add(text);
459 break;
460 case 5: // link
461 Link link = instructionalFactory.createLink();
462 link.setUrl(rs.getString("url"));
463 link.setLabel(rs.getString("description"));
464 link.setDifficultyLevel(rs.getInt("difficultyLevel"));
465 link.setLearningStyle(rs.getInt("learningStyle"));
466 link.setVisible(rs.getBoolean("visible"));
467 listMobileMedia.add(link);
468 break;
469 default:
470 break;
471 }
472 }
```

```
473     return listMobileMedia;
474 }
475
476 private Module getModuleDB(int idmodule) throws SQLException {
477     ResultSet rs = getResultSetAll("*", "\"Module\"", "\"idModule\"", idmodule,
478     "\"idModule\"");
479     return createModule(rs);
480 }
481
482 private Module createModule(ResultSet rs) throws SQLException {
483     Module obj = instructionalFactory.createModule();
484     while (rs.next()) {
485         obj.setLabel(rs.getString("nameModule"));
486         obj.setNameModule(rs.getString("nameModule"));
487         obj.setVisible(rs.getBoolean("visible"));
488     }
489     return obj;
490 }
491
492 public static void saveModel(IKnowledgeDomain domain, String arquivo) {
493     IKnowledgeDomain knowledgeDomain = instructionalFactory.
494     createIKnowledgeDomain();
495     knowledgeDomain = domain;
496     // Obtendo novo resource set
497     ResourceSet resSet = new ResourceSetImpl();
498     resSet.getResourceFactoryRegistry().getExtensionToFactoryMap().put("
499     mminstructional",
500     new XMIResourceFactoryImpl());
501     // arquivo = arquivo+".xmi";
502     // Create resource
503     System.out.println("Path: " + arquivo + "\n");
504     URI fileURI = URI.createURI(arquivo + ".mminstructional");
505     // URI.decode("UTF-8");
506     Resource resource = resSet.createResource(fileURI);
507     resource.getContents().add(knowledgeDomain);
508     try {
509         resource.save(null);
510         System.out.println("*** Save *** \n" + fileURI.toString());
511     } catch (IOException e) {
512         System.out.println("erro salvar modelo...");
513         e.printStackTrace();
514     }
515 }
516
517 /**
518  * @return the listModules
519  */
520 public BasicEList<Module> getListModules() {
521     return listModules;
522 }
523
524 /**
525  * @param listModules the listModules to set
526  */
527 public void setListModules(BasicEList<Module> listModules) {
528     this.listModules = listModules;
529 }
```

```
527
528 /**
529  * @return the listConcepts
530  */
531 public BasicEList<IConcept> getListConcepts() {
532     return listConcepts;
533 }
534
535 /**
536  * @param listConcepts the listConcepts to set
537  */
538 public void setListConcepts(BasicEList<IConcept> listConcepts) {
539     this.listConcepts = listConcepts;
540 }
541
542 /**
543  * @return the domain
544  */
545 public IKnowledgeDomain getDomain() {
546     return domain;
547 }
548
549 /**
550  * @param domain the domain to set
551  */
552 public void setDomain(IKnowledgeDomain domain) {
553     this.domain = domain;
554 }
555
556 }
```

A.2 Instância do Modelo Mobile

O Modelo Mobile é construído a partir da aplicação de regras de transformação no Modelo Instrucional. A regra de transformação a seguir cria uma instância do Modelo Mobile.

Código-fonte 3 – Regras de transformação em Java - Modelo Educom para Modelo Mobile

```
1 package instrucionalToMobileLearning;
2 import java.io.IOException;
3 import org.eclipse.emf.common.util.BasicEList;
4 import org.eclipse.emf.common.util.EList;
5 import org.eclipse.emf.common.util.URI;
6 import org.eclipse.emf.ecore.resource.Resource;
7 import org.eclipse.emf.ecore.resource.ResourceSet;
8 import org.eclipse.emf.ecore.resource.impl.ResourceSetImpl;
9 import org.eclipse.emf.ecore.xmi.impl.XMIResourceFactoryImpl;
10 import mmInstructional.AnswersAlternatives;
11 import mmInstructional.CollaborativeActivity;
12 import mmInstructional.Concept;
13 import mmInstructional.IConcept;
14 import mmInstructional.IKnowledgeDomain;
15 import mmInstructional.InformationItem;
16 import mmInstructional.InstrucElementType;
```

```
17 import mmInstructional.InstructionalElement;
18 import mmInstructional.Link;
19 import mmInstructional.MmInstructionalFactory;
20 import mmInstructional.MmInstructionalPackage;
21 import mmInstructional.MobileMedia;
22 import mmInstructional.Module;
23 import mmInstructional.Phase;
24 import mmInstructional.Procedure;
25 import mmInstructional.Question;
26 import mmMobile.Audio;
27 import mmMobile.ElementInterface;
28 import mmMobile.Image;
29 import mmMobile.MConcept;
30 import mmMobile.MKnowledgeDomain;
31 import mmMobile.MModule;
32 import mmMobile.MmMobileFactory;
33 import mmMobile.MmMobilePackage;
34 import mmMobile.Screen;
35 import mmMobile.Text;
36 import mmMobile.Video;
37
38 public class Educomm1ToMobile {
39     static MmMobileFactory mobileFactory = MmMobileFactory.eINSTANCE;
40     static MmInstructionalFactory instructionalFactory = MmInstructionalFactory.
41         eINSTANCE;
42     MmMobilePackage mobilePackage = MmMobilePackage.eINSTANCE;
43     MmInstructionalPackage instructionalPackage = MmInstructionalPackage.eINSTANCE
44         ;
45     MKnowledgeDomain domainMobile = mobileFactory.createMKnowledgeDomain();
46     IKnowledgeDomain ikDomain = instructionalFactory.createIKnowledgeDomain();
47     private String pathFile;
48     private String nameFile;
49     private Screen screenModuleDomain;
50     public Educomm1ToMobile() {
51     }
52     public void transformation() {
53         ikDomain = loadModel(getPathFile() + getNameFile() + ".mminstructional");
54         createMKnowledgeDomain(ikDomain);
55         // Carregar conceitos do Modelo Instructional
56         if (ikDomain.getIHasConcepts().size() > 0) {
57             ikDomain.getIHasConcepts().forEach(iconcept -> {
58                 MConcept mConcept = createMConcept(iconcept);
59                 domainMobile.getMHasConcept().add(mConcept);
60             });
61         }
62         screenModuleDomain = mobileFactory.createScreen();
63         if (!ikDomain.getIHasModule().isEmpty()) {
64             EList<Module> iModules = ikDomain.getIHasModule();
65             // Dominio tem modulos?
66             ikDomain.getIHasModule().forEach(iModule -> {
67                 MModule mModule = createMModule(iModule);
68                 screenModuleDomain.getScreenToMModule().add(mModule);
69                 if (!iModule.getModuleToIElement().isEmpty()) {
70                     EList<InstructionalElement> iElements = iModule.getModuleToIElement();
71                     for (InstructionalElement iElement : iElements) {
72                         EList<CollaborativeActivity> collaborativeActivitys = iElement.
```

```

getIElementCollab();
72     for (CollaborativeActivity cActivity : collaborativeActivitys) {
73         Screen screen = mobileFactory.createScreen();
74         screen = screenCollaborativeActivity(cActivity);
75         screen.setLabel(iElement.getHasIElementType().getName());
76         screen.setLabel(cActivity.getLabel());
77         screen.setOrigemElement(cActivity.getClass().getSimpleName());
78         mModule.getMModuleToScreen().add(screen);
79     }
80 }
81 }
82 });
83     domainMobile.getMKdomainToScreen().add(screenModuleDomain);
84 }
85     saveMobileModel(domainMobile, getPathFile() + getNameFile());
86
87     protected EList<Screen> questionScreen(InstructionalElement instElement) {
88         EList<Question> questions = instElement.getIElementQuestion();
89         EList<Screen> screens = new BasicEList<Screen>();
90         for (Question question : questions) {
91             screens.add(screenQuestion(question));
92         }
93         return screens;
94     }
95
96     protected ElementInterface instructionalElement2ElementInterface(
97         InstructionalElement iElement) {
98         InstrucElementType elementType = iElement.getHasIElementType();
99         ElementInterface elementInterface = mobileFactory.createText();
100        String objIE = "Exercicio";
101        if (objIE.equals(iElement.getHasIElementType().getName())) {
102            if (!iElement.getIElementCollab().isEmpty()) {
103                for (CollaborativeActivity c : iElement.getIElementCollab()) {
104                    for (MobileMedia m : c.getColActivityToMMedia()) {
105                        elementInterface = mobileMediaToElementoInterface(m);
106                    }
107                }
108            }
109            if (!iElement.getIElementQuestion().isEmpty()) {
110                EList<Question> questions = iElement.getIElementQuestion();
111                for (Question question : questions) {
112                    }
113            }
114            if (!iElement.getIElementMMedia().isEmpty()) {
115                for (MobileMedia m : iElement.getIElementMMedia()) {
116                    elementInterface = mobileMediaToElementoInterface(m);
117                }
118            }
119        }
120        objIE = "Link";
121        if (objIE.equals(elementType.getName())) {
122            for (MobileMedia m : iElement.getIElementMMedia()) {
123                elementInterface = mobileMediaToElementoInterface(m);
124            }
125        }
126        objIE = "Collaborative";

```



```

127     if (objIE.equals(elementType.getName())) {
128         for (MobileMedia m : iElement.getIElementMMedia()) {
129             elementInterface = mobileMediaToElementoInterface(m);
130         }
131     }
132     return elementInterface;
133 }
134
135 private ElementInterface createQuestionElementInterface(Question question) {
136     Text mText = mobileFactory.createText();
137     mText.setLabel(question.getDescriptionQuestion());
138     mText.setTextDisplay(question.getOrderQuestion() + ") " + question.
139     getDescriptionQuestion());
140     mText.setOrigemElement(mText.getClass().getSimpleName());
141     return mText;
142 }
143
144 public EducommlToMobile(String pathFile, String nameFile) {
145     this.setPathFile(pathFile);
146     this.setNameFile(nameFile);
147 }
148
149 private Audio createAudio(mmInstructional.Audio iAudio) {
150     Audio mAudio = mobileFactory.createAudio();
151     mAudio.setLabel(iAudio.getLabel());
152     mAudio.setPathAudio(iAudio.getPath());
153     mAudio.setNameAudio(iAudio.getNameFile());
154     iAudio.getSubMMedia().forEach(iSubMobileMedia -> {
155         mAudio.getSubElementInterface().add(mobileMediaToElementoInterface(
156         iSubMobileMedia));
157     });
158     return mAudio;
159 }
160
161 private Screen screenCollaborativeActivity(CollaborativeActivity
162 collaborativeActivity) {
163     Screen collaborativeScreen = mobileFactory.createScreen();
164     Text mText = mobileFactory.createText();
165     collaborativeScreen.getScreenToElementInterface()
166     .addAll(CollaborativeActivityToElementInterface(collaborativeActivity));
167     return collaborativeScreen;
168 }
169
170 private EList<ElementInterface> CollaborativeActivityToElementInterface(
171 CollaborativeActivity collaborativeActivity) {
172     EList<ElementInterface> elementInterface = new BasicEList<ElementInterface
173 >();
174     if (!collaborativeActivity.getColActivityToMMedia().isEmpty()) {
175         for (MobileMedia m : collaborativeActivity.getColActivityToMMedia()) {
176             elementInterface.add(mobileMediaToElementoInterface(m));
177         }
178     }
179     return elementInterface;
180 }
181
182 private EList<ElementInterface> createElementInterfaceInformationItem(EList<

```

```

InformationItem> informationItems) {
180  EList<ElementInterface> elementInterfaces = new BasicEList<ElementInterface
    >();
181  informationItems.forEach(informationItem -> {
182      if (informationItem instanceof Concept) {
183          Concept concept = (Concept) informationItem;
184          concept.getConceptHasMMedia().forEach(mobileMedia -> {
185              ElementInterface elementInterface = mobileMediaToElementoInterface(
                mobileMedia);
186              if (elementInterface != null) {
187                  elementInterfaces.add(elementInterface);
188              } else {
189                  System.out.println("elemento interface nulo");
190              }
191          });
192      } else if (informationItem instanceof Procedure) {
193          Procedure procedure = (Procedure) informationItem;
194          procedure.getProcToMMedia().forEach(mobileMedia -> {
195              ElementInterface elementInterface = mobileMediaToElementoInterface(
                mobileMedia);
196              if (elementInterface != null) {
197                  elementInterfaces.add(elementInterface);
198              } else {
199                  System.out.println("elemento interface nulo");
200              }
201          });
202      });
203
204      procedure.getPhase().forEach(phases -> {
205          Text mText = createTextPhase(phases);
206          elementInterfaces.add(mText);
207      });
208  }
209 });
210 return elementInterfaces;
211 }
212
213 public Image createImage(mmInstructional.Image image) {
214     Image mImage = mobileFactory.createImage();
215     mImage.setLabel(image.getLabel());
216     mImage.setNameImg(image.getNameFile());
217     mImage.setOrigemElement(image.getClass().getSimpleName());
218     mImage.setPathImg(image.getPath());
219     mImage.setTextInformation(image.getDescription());
220     image.getSubMMedia().forEach(iSubMobileMedia -> {
221         mImage.getSubElementInterface().add(mobileMediaToElementoInterface(
            iSubMobileMedia));
222     });
223     return mImage;
224 }
225
226
227 private mmMobile.Link createLink(Link iLink) {
228     mmMobile.Link mLink = mobileFactory.createLink();
229     mLink.setLabel(iLink.getLabel());
230     mLink.setLink(iLink.getUrl());
231     mLink.setOrigemElement(iLink.getClass().getSimpleName());

```

```

232     iLink.getSubMMedia().forEach(iSubMobileMedia -> {
233         mLink.getSubElementInterface().add(mobileMediaToElementoInterface(
234             iSubMobileMedia));
235     });
236     return mLink;
237 }
238
239 public MConcept createMConcept(IConcept iconcepts) {
240     MConcept mConcept = mobileFactory.createMConcept();
241     mConcept.setNameMConcept(iconcepts.getNameIConcept());
242     EList<ElementInterface> elementInterface =
243     createElementInterfaceInformationItem(iconcepts.getHasInforItem());
244     mConcept.getMConceptToElementInterface().addAll(elementInterface);
245     iconcepts.getHasInstrucinfor().forEach(instructionalElement -> {
246         if (instructionalElement.getHasIElementType().getName().equals("Exemplo"))
247         {
248             instructionalElement.getIElementMMedia().forEach(mobileMedia -> {
249                 mConcept.getMConceptToElementInterface().add(
250                 mobileMediaToElementoInterface(mobileMedia));
251             });
252         } else {
253             mConcept.getMConceptToElementInterface()
254             .add(instructionalElement2ElementInterface(instructionalElement));
255         }
256     });
257     iconcepts.getIsubConcept().forEach(iSubConcept -> {
258         mConcept.getMSubConcept().add(createMConcept(iSubConcept));
259     });
260     return mConcept;
261 }
262
263 public void createMKnowledgeDomain(IKnowledgeDomain iKnowledgeDomain) {
264     domainMobile.setMNameKD(iKnowledgeDomain.getNameIKDomain());
265 }
266
267 private MModule createMModule(Module iModule) {
268     MModule mModule = mobileFactory.createMModule();
269     mModule.setLabel(iModule.getLabel());
270     mModule.setNameModule(iModule.getNameModule());
271     mModule.setVisible(true);
272     Screen screenMConcept = mobileFactory.createScreen();
273     if (iModule.getModuleToMMedia().size() > 0) {
274         Screen screenEInterface = mobileFactory.createScreen();
275         for (MobileMedia mm : iModule.getModuleToMMedia()) {
276             screenEInterface.getScreenToElementInterface().add(
277             mobileMediaToElementoInterface(mm));
278         }
279         mModule.getMModuleToScreen().add(screenEInterface);
280     }
281     // Modulo tem conceitos?
282     if (!iModule.getIModuleHasConcepts().isEmpty()) {
283         screenMConcept.setLabel(iModule.getNameModule());
284         screenMConcept.setOrigemElement(iModule.getClass().getSimpleName());
285         iModule.getIModuleHasConcepts().forEach(iConcept -> {
286             System.out.println((iConcept.getNameIConcept()));
287             MConcept mConcept = mobileFactory.createMConcept();
288             mConcept = createMConcept(iConcept);

```

```

284     screenMConcept.getScreenToMConcept().add(mConcept);
285     EList<InstructionalElement> iElements = new BasicEList<
InstructionalElement>();
286     iElements = iConcept.getHasInstrucinfor();
287     for (InstructionalElement iElement : iElements) {
288         Screen screenIEMConcept = mobileFactory.createScreen();
289         if (!iElement.getIElementQuestion().isEmpty()) {
290             EList<Question> questions = iElement.getIElementQuestion();
291             for (Question question : questions) {
292                 Screen screenQuestion = mobileFactory.createScreen();
293                 screenQuestion = screenQuestion(question);
294                 screenQuestion.setLabel(iElement.getHasIElementType().getName());
295                 mConcept.getMConceptToScreen().add(screenQuestion);
296             }
297         }
298     }
299 }
300 });
301 mModule.getMModuleToScreen().add(screenMConcept);
302 }
303
304 // Tem instructional Element?
305
306 if (!iModule.getModuleToIElement().isEmpty()) {
307     for (InstructionalElement iE : iModule.getModuleToIElement()) {
308         String obj = "Exercício";
309         if (iE.getHasIElementType().getName().equals(obj)) {
310             Screen screenMModuleIE = mobileFactory.createScreen();
311             if (!iE.getIElementQuestion().isEmpty()) {
312                 EList<Question> questions = iE.getIElementQuestion();
313                 for (Question question : questions) {
314                     screenMModuleIE = screenQuestion(question);
315                     screenMModuleIE.setLabel(iE.getHasIElementType().getName());
316                     screenMModuleIE.setTypeAvaliation(1);
317                     mModule.getMModuleToScreen().add(screenMModuleIE);
318                 }
319             }
320         }
321         String objLink = "Link";
322         if (objLink.equals(iE.getHasIElementType().getName())) {
323             for (MobileMedia mb : iE.getIElementMMedia()) {
324                 screenMConcept.getScreenToElementInterface().add(
mobileMediaToElementoInterface(mb));
325             }
326         }
327     }
328 }
329 // Modulo tem sub-modulos?
330 if (!iModule.getSubModule().isEmpty()) {
331     Screen mSubModuleScreen = mobileFactory.createScreen();
332     iModule.getSubModule().forEach(isubmodule -> {
333         createScreenModule(iModule, mSubModuleScreen);
334         mSubModuleScreen.getScreenToMModule().add(createMModule(isubmodule));
335     });
336     mModule.getMModuleToScreen().add(mSubModuleScreen);
337 }
338 return mModule;

```

```
339 }
340
341 private ElementInterface createAnswersAlternatives(AnswersAlternatives answer)
342 {
343     Text mText = mobileFactory.createText();
344     mText.setLabel(answer.getAnswers());
345     mText.setTextDisplay(answer.getIdObjAnswer() + ") " + answer.getAnswers());
346     mText.setOrigemElement(answer.getClass().getSimpleName());
347     if (answer.getIsTrue()) {
348     }
349     return mText;
350 }
351
352 protected Screen screenInstructionalElement(InstructionalElement iElement) {
353     Screen screen = mobileFactory.createScreen();
354     if (!iElement.getIElementQuestion().isEmpty()) {
355         EList<Question> questionList = iElement.getIElementQuestion();
356         for (Question question : questionList) {
357             screen = screenQuestion(question);
358         }
359     }
360     if (!iElement.getIElementCollab().isEmpty()) {
361         EList<CollaborativeActivity> collab = iElement.getIElementCollab();
362         screen.setLabel(iElement.getLabel());
363         for (CollaborativeActivity collaborativeActivity : collab) {
364             screen = screenCollaborativeActivity(collaborativeActivity);
365         }
366     }
367     if (!iElement.getIElementMMedia().isEmpty()) {
368         EList<MobileMedia> mobileMedia = iElement.getIElementMMedia();
369         for (MobileMedia m : mobileMedia) {
370             System.out.println("}}}" + m.getLabel());
371             screen.setLabel(iElement.getLabel());
372             screen.setOrigemElement(iElement.getClass().getName());
373             screen.getScreenToElementInterface().add(mobileMediaToElementInterface(
374             m));
375         }
376     }
377     return screen;
378 }
379
380 private Screen screenQuestion(Question question) {
381     Screen screenQuestion = mobileFactory.createScreen();
382     Text mText = mobileFactory.createText();
383     mText.setLabel(question.getDescriptionQuestion());
384     mText.setOrigemElement("Question");
385     mText.setTextDisplay(question.getOrderQuestion() + ") " + question.
386     getDescriptionQuestion());
387     int index = 1;
388     for (AnswersAlternatives iAnswer : question.getQuestionToAnswers()) {
389         Text mTextAnswers = mobileFactory.createText();
390         mTextAnswers.setLabel(iAnswer.getAnswers());
391         mTextAnswers.setTextDisplay(iAnswer.getIdObjAnswer() + ") "+ iAnswer.
392         getAnswers());
393         mTextAnswers.setOrigemElement(mTextAnswers.getClass().getSimpleName());
394         mTextAnswers.setIsRightAnswer(iAnswer.getIsTrue());
```

```
392     if (!iAnswer.getIsTrue()) {
393         index++;
394     }
395     mText.getSubElementInterface().add(mTextAnswers);
396 }
397 screenQuestion.setLabel(mText.getLabel());
398 screenQuestion.setOrigemElement(question.getClass().getSimpleName());
399 screenQuestion.setTypeAvaliation(question.getTypeQuestion());
400 screenQuestion.getScreenToElementInterface().add(mText);
401 return screenQuestion;
402 }
403
404 protected void createScreenModule(Module iModule, Screen screen) {
405     screen.setLabel(iModule.getNameModule());
406     screen.setOrigemElement(iModule.getNameModule());
407 }
408
409 public Text createText(mmInstructional.Text text) {
410     Text mText = mobileFactory.createText();
411     mText.setLabel(text.getLabel());
412     mText.setTextDisplay(text.getTextFull());
413     mText.setOrigemElement(text.getClass().getSimpleName());
414     text.getSubMMedia().forEach(iSubMobileMedia -> {
415         mText.getSubElementInterface().add(mobileMediaToElementoInterface(
416             iSubMobileMedia));
417     });
418     return mText;
419 }
420
421 protected Text createTextPhase(Phase phase) {
422     Text mText = mobileFactory.createText();
423     mText.setLabel(phase.getDescription());
424     mText.setTextDisplay(phase.getDescription());
425     mText.setOrigemElement(mText.getClass().getSimpleName());
426     phase.getPhaseToMMedia().forEach(mobileMedia -> {
427         mText.getSubElementInterface().add(mobileMediaToElementoInterface(
428             mobileMedia));
429     });
430     return mText;
431 }
432
433 private Video createVideo(mmInstructional.Video iVideo) {
434     Video mVideo = mobileFactory.createVideo();
435     mVideo.setLabel(iVideo.getLabel());
436     mVideo.setNameVideo(iVideo.getNameFile());
437     mVideo.setPathVideo(iVideo.getPath());
438     mVideo.setTime((int) iVideo.getTime());
439     mVideo.setOrigemElement(iVideo.getClass().getSimpleName());
440     iVideo.getSubMMedia().forEach(iSubMobileMedia -> {
441         mVideo.getSubElementInterface().add(mobileMediaToElementoInterface(
442             iSubMobileMedia));
443     });
444     return mVideo;
445 }
446
447 private String getNameFile() {
448     return nameFile;
449 }
```

```
446 }
447 private String getPathFile() {
448     return pathFile;
449 }
450 private ElementInterface mobileMediaToElementoInterface(MobileMedia
mobileMedia) {
451     ElementInterface elementInterface = mobileFactory.createElementInterface();
452     if (mobileMedia.isVisible()) {
453         if (mobileMedia instanceof mmInstructional.Text) {
454             Text mText = mobileFactory.createText();
455             mmInstructional.Text text = (mmInstructional.Text) mobileMedia;
456             mText = createText(text);
457             elementInterface = mText;
458         } else {
459             if (mobileMedia instanceof mmInstructional.Image) {
460                 Image mImage = mobileFactory.createImage();
461                 mImage = createImage((mmInstructional.Image) mobileMedia);
462                 elementInterface = mImage;
463             } else {
464                 if (mobileMedia instanceof Link) {
465                     mmMobile.Link mLink = mobileFactory.createLink();
466                     mLink = createLink((Link) mobileMedia);
467                     elementInterface = mLink;
468                 } else {
469                     if (mobileMedia instanceof mmInstructional.Audio) {
470                         Audio mAudio = mobileFactory.createAudio();
471                         mAudio = createAudio((mmInstructional.Audio) mobileMedia);
472                         elementInterface = mAudio;
473                     } else {
474                         if (mobileMedia instanceof mmInstructional.Video) {
475                             Video mVideo = mobileFactory.createVideo();
476                             mVideo = createVideo((mmInstructional.Video) mobileMedia);
477                             elementInterface = mVideo;
478                         }
479                     }
480                 }
481             }
482         }
483     }
484     return elementInterface;
485 }
486
487 private void setNameFile(String nameFile) {
488     this.nameFile = nameFile;
489 }
490
491 private void setPathFile(String pathFile) {
492     this.pathFile = pathFile;
493 }
494
495 private static IKnowledgeDomain loadModel(String loadFile) {
496     IKnowledgeDomain ickDomain = instructionalFactory.createIKnowledgeDomain();
497     ResourceSet resourceSet = new ResourceSetImpl();
498     resourceSet.getResourceFactoryRegistry().getExtensionToFactoryMap().put("
mminstructional",
499         new XMIResourceFactoryImpl());
500     URI fileURI = URI.createURI(loadFile);
```

```

501 Resource resource = resourceSet.getResource(fileURI, true);
502 try {
503     resource.load(null);
504     ickDomain = (IKnowledgeDomain) resource.getContents().get(0);
505     // resource.save(System.out, Collections.EMPTY_MAP);
506     return ickDomain;
507 } catch (IOException e) {
508     System.out.println("failed to read " + fileURI);
509 }
510 return null;
511 }
512
513 public static void saveMobileModel(MKnowledgeDomain mKnowledgeDomain, String
    arquivo) {
514     MKnowledgeDomain knowledgeDomain = mobileFactory.createMKnowledgeDomain();
515     knowledgeDomain = mKnowledgeDomain;
516     // Obtendo novo resource set
517     ResourceSet resSet = new ResourceSetImpl();
518     resSet.getResourceFactoryRegistry().getExtensionToFactoryMap().put("mmMobile
", new XMIResourceFactoryImpl());
519     // arquivo = arquivo+"xmi";
520     // Create resource
521     URI fileURI = URI.createURI(arquivo + ".mmMobile");
522     // URI.decode("UTF-8");
523     Resource resource = resSet.createResource(fileURI);
524     resource.getContents().add(knowledgeDomain);
525     try {
526         resource.save(null);
527         System.out.println("Path: " + fileURI.toFileString());
528         System.out.println("Salvo...");
529     } catch (IOException e) {
530         System.out.println("erro salvar modelo mobile ...");
531         e.printStackTrace();
532     }
533 }
534
535 }

```

A.3 Instância do Modelo React Native

O Modelo React Native é construído a partir da aplicação de regras de transformação no Modelo Mobile. A regra de transformação a seguir cria uma instância do Modelo React Native.

Código-fonte 4 – Regras de transformação em Java - Modelo Mobile para Modelo React-Native

```

1 package mm.mobileLearning.emf.implement;
2 import java.util.ArrayList;
3 import java.util.List;
4 import mmMobile.Link;
5 import mmMobile.MmMobileFactory;
6 import mmMobileLearning.Image;
7 import mmMobileLearning.MModule;
8 import mmMobileLearning.Screen;
9 import mmMobileLearning.Text;

```



```

65         if (elementModule instanceof Text) {
66             ObjectElement objectElementesModules = createObjectElement(
Boolean.TRUE);
67             arrayElementModules.getArrayToObjects().add(
objectElementesModules);
68             objectElementModule.getObjectToArray().add(arrayElementModules);
69             ElementText elementTextExerciceContents = createElementText("
text",
70                 text.getTextDisplay());
71             objectElementesModules.getElementText().add(
elementTextExerciceContents);
72             Array arraySubElement = createArray("exercices");
73             text.getSubElementInterface().forEach(subElement -> {
74                 objectElementesModules.getObjectToArray().add(arraySubElement)
;
75                 ObjectElement objectElementSub = createObjectElement(Boolean.
TRUE);
76                 arraySubElement.getArrayToObjects().add(objectElementSub);
77                 if (subElement instanceof Text) {
78                     Text subText = ((Text) subElement);
79                     ElementText subElementTextModulesContents =
createElementText("subText",
80                         subText.getTextDisplay());
81                     objectElementSub.getElementText().add(
subElementTextModulesContents);
82                 }
83             });
84         }
85     }
86 });
87 });
88
89     module.getMModuleToMConcept().forEach(concept -> {
90         ObjectElement objectElementConcept = createObjectElement(Boolean.TRUE)
;
91         ElementText elementTextConcept = createElementText("concept", concept.
getNameMConcept());
92         objectElementConcept.getElementText().add(elementTextConcept);
93         arrayConcept.getArrayToObjects().add(objectElementConcept);
94         elementTextModule.getElementTextToArray().add(arrayConcept);
95         // Criar array dos elementos concepts
96         concept.getMConceptToScreen().forEach(screenConcept -> {
97             Array arrayElements = createArray(screenConcept.getLabel());
98             elementTextConcept.getElementTextToArray().add(arrayElements);
99             screenConcept.getScreenToElementInterface().forEach(elementConcept
-> {
100                 if (elementConcept instanceof Text) {
101                     ObjectElement objectElements = createObjectElement(Boolean.FALSE
);
102                     arrayElements.getArrayToObjects().add(objectElements);
103                     Text text = ((Text) elementConcept);
104                     if (screenConcept.getLabel().contains("IConceptImpl")) {
105                         ElementText elementTextConceptsContents = createElementText("
text",
106                             text.getTextDisplay());
107                         objectElements.getElementText().add(
elementTextConceptsContents);

```

```

108         Array arraySubElement = createArray("subTexts");
109         text.getSubElementInterface().forEach(subElement -> {
110             objectElements.getObjectToArray().add(arraySubElement);
111             ObjectElement objectElementSub = createObjectElement(Boolean
112                 .TRUE);
113             arraySubElement.getArrayToObjects().add(objectElementSub);
114             if (subElement instanceof Text) {
115                 Text subText = ((Text) subElement);
116                 ElementText subElementTextModulesContents =
117                 createElementText("subText",
118                     subText.getTextDisplay());
119                 objectElementSub.getElementText().add(
120                 subElementTextModulesContents);
121             }
122         });
123     }
124
125     if (screenConcept.getLabel().contains("exercise")) {
126         ElementText elementTextExerciceContents = createElementText("
127         text",
128             text.getTextDisplay());
129         objectElements.getElementText().add(
130         elementTextExerciceContents);
131         Array arraySubElement = createArray("exercices");
132         text.getSubElementInterface().forEach(subElement -> {
133             objectElements.getObjectToArray().add(arraySubElement);
134             ObjectElement objectElementSub = createObjectElement(Boolean
135                 .TRUE);
136             arraySubElement.getArrayToObjects().add(objectElementSub);
137             if (subElement instanceof Text) {
138                 Text subText = ((Text) subElement);
139                 ElementText subElementTextModulesContents =
140                 createElementText("subText",
141                     subText.getTextDisplay());
142                 objectElementSub.getElementText().add(
143                 subElementTextModulesContents);
144             }
145         });
146     }
147
148     if (elementConcept instanceof Image) {
149         ObjectElement objectElements = createObjectElement(Boolean.FALSE
150         );
151         arrayElements.getArrayToObjects().add(objectElements);
152         Image image = ((Image) elementConcept);
153         if (screenConcept.getLabel().contains("IConceptImpl")) {
154             ElementImage elementImageConceptsContents = createElementImage
155             (image.getPathImg(),
156                 image.getNameImg(), "image");
157             objectElements.getElementImage().add(
158             elementImageConceptsContents);
159             Array arraySubElement = createArray("subImages");
160             image.getSubElementInterface().forEach(subElement -> {
161                 objectElements.getObjectToArray().add(arraySubElement);
162                 ObjectElement objectElementSub = createObjectElement(Boolean
163                     .TRUE);
164                 arraySubElement.getArrayToObjects().add(objectElementSub);

```

```

153         if (subElement instanceof Image) {
154             Image subImage = ((Image) subElement);
155             ElementImage subElementImageModulesContents =
createElementImage(
156                 subImage.getPathImg(), subImage.getNameImg(), "
subImage");
157                 objectElementSub.getElementImage().add(
subElementImageModulesContents);
158             }
159         });
160     }
161     if (screenConcept.getLabel().contains("exercise")) {
162         ElementImage elementImageExerciceContents = createElementImage
(image.getPathImg(),
163             image.getNameImg(), "image");
164         objectElements.getElementImage().add(
elementImageExerciceContents);
165         Array arraySubElement = createArray("exercices");
166         image.getSubElementInterface().forEach(subElement -> {
167             objectElements.getObjectToArray().add(arraySubElement);
168             ObjectElement objectElementSub = createObjectElement(Boolean
.TRUE);
169             arraySubElement.getArrayToObjects().add(objectElementSub);
170             if (subElement instanceof Image) {
171                 Image subImage = ((Image) subElement);
172                 ElementImage subElementImageModulesContents =
createElementImage(
173                     subImage.getPathImg(), subImage.getNameImg(), "image")
;
174                     objectElementSub.getElementImage().add(
subElementImageModulesContents);
175                 }
176             });
177         }
178     }
179     if (elementConcept instanceof Link) {
180         ObjectElement objectElements = createObjectElement(Boolean.FALSE
);
181         Link link = ((Link) elementConcept);
182         ElementText elementTextConceptContents = createElementText("link
", link.getLink());
183         objectElements.getElementText().add(elementTextConceptContents);
184         link.getSubElementInterface().forEach(subElement -> {
185             Text subText = ((Text) subElement);
186             ElementText subElementTextModulesContents = createElementText(
"text",
187                 subText.getTextDisplay());
188             objectElements.getElementText().add(
subElementTextModulesContents);
189         });
190     }
191     if (elementConcept instanceof Image) {
192         ObjectElement objectElements = createObjectElement(Boolean.FALSE
);
193         Image image = ((Image) elementConcept);
194         ElementImage elementTextModulesContents = createElementImage(
image.getPathImg(),

```

```

195         "image", "image");
196         objectElements.getElementImage().add(elementTextModulesContents)
197         ;
198         image.getSubElementInterface().forEach(subElement -> {
199             Text subText = ((Text) subElement);
200             ElementText subElementTextModulesContents = createElementText(
201                 "text",
202                 subText.getTextDisplay());
203                 objectElements.getElementText().add(
204                 subElementTextModulesContents);
205             });
206         });
207     });
208     });
209
210     this.projetoRN.getComponents().add(component);
211     this.projetoRN.setProjectToData(dataJSModule);
212     return this.projetoRN;
213 }
214
215 private DataJS createDataJS(String name) {
216     DataJS dataJS = reactNativeFactory.createDataJS();
217     dataJS.setName(name);
218     return dataJS;
219 }
220
221 private StyleSheet createStyleSheet(String nameStyle, Border border, FlexBox
222     flexBox, Padding padding,
223     Margin margin, Font font) {
224     StyleSheet styleSheet = reactNativeFactory.createStyleSheet();
225     styleSheet.setNameStyle(nameStyle);
226     styleSheet.setHasBorder(border);
227     styleSheet.setHasFlexbox(flexBox);
228     styleSheet.setHasFont(font);
229     styleSheet.setHasPadding(padding);
230     styleSheet.setHasMargin(margin);
231     return styleSheet;
232 }
233
234 private ElementText createElementText(String value, String content) {
235     ElementText elementText = reactNativeFactory.createElementText();
236     elementText.setValueText(value);
237     elementText.setContentText(content);
238     return elementText;
239 }
240
241 private Component createComponent(String NameComponent, Boolean isState) {
242     Component component = reactNativeFactory.createComponent();
243     component.setComponentName(NameComponent);
244     component.setState(isState);
245     return component;
246 }
247
248 private ElementImage createElementImage(String path, String contentImage,

```

```
String valueImage) {
248     ElementImage elementImage = reactNativeFactory.createElementImage();
249     elementImage.setContentImage(path.concat(contentImage));
250     elementImage.setValueImage(valueImage);
251     return elementImage;
252 }
253
254 private ObjectElement createObjectElement(Boolean isId) {
255     ObjectElement objectElement = reactNativeFactory.createObjectElement();
256     objectElement.setId(isId);
257     return objectElement;
258 }
259
260 private Array createArray(String nameArray) {
261     Array array = reactNativeFactory.createArray();
262     array.setValue(nameArray);
263     return array;
264 }
265
266 private FlatList createFlatList(String data, String flatListName, String
renderItem, String keyExtractor) {
267     FlatList flatList = reactNativeFactory.createFlatList();
268     flatList.setData(data);
269     flatList.setFlatlistName(flatListName);
270     flatList.setKeyExtractor(keyExtractor);
271     flatList.setRenderItem(renderItem);
272     return flatList;
273 }
274
275 private View createView(String viewName, Integer viewType) {
276     View view = reactNativeFactory.createView();
277     view.setViewName(viewName);
278     view.setViewType(viewType);
279     return view;
280 }
281
282 private View getViewToFlatList(View view, Screen screen) {
283     FlatList flatList;
284     if (view.getViewName().equals("module")) {
285         flatList = reactNativeFactory.createFlatList();
286         flatList.setFlatlistName("module");
287         view.getViewToflatlists().add(getFlatList(flatList, screen));
288     }
289     return view;
290 }
291
292 private FlatList getFlatList(FlatList flatList, Screen screen) {
293     Array array;
294     if (flatList.getFlatlistName().equals("module")) {
295         array = reactNativeFactory.createArray();
296         array.setValue(flatList.getFlatlistName());
297         flatList.getFlatListToArray().add(getArrayAndScreen(array, screen));
298     }
299     return flatList;
300 }
301
302 private Array getArrayAndScreen(Array array, Screen screen) {
```

```

303     ObjectElement objectElement;
304     if (array.getValue().equals("module") || array.getValue().equals("concepts")
305     ) {
306         objectElement = reactNativeFactory.createObjectElement();
307         array.getArrayToObjects().add(getObjectElementAndScreen(objectElement,
308         screen));
309     }
310     return array;
311 }
312 private ObjectElement getObjectElementAndScreen(ObjectElement objectElement,
313     Screen screen) {
314     this.array = reactNativeFactory.createArray();
315     this.objectElement = reactNativeFactory.createObjectElement();
316     screen.getScreenToMModule().forEach(module -> {
317         ElementText elementTextModule = this.reactNativeFactory.createElementText
318         ();
319         elementTextModule.setValueText(module.getLabel());
320         elementTextModule.setContentText(module.getNameModule());
321         module.getMModuleToMConcept().forEach(concept -> {
322             ElementText elementTextConcept = reactNativeFactory.createElementText();
323             this.array.setValue("concepts");
324             this.array.getArrayToObjects().add(this.objectElement);
325             elementTextConcept.getElementTextToArray().add(this.array);
326             elementTextConcept.setValueText("concept");
327             elementTextConcept.setContentText(concept.getNameMConcept());
328             this.objectElement.getElementText().add(elementTextConcept);
329             objectElement.getObjectToArray().add(array);
330             elementTextModule.getElementTextToArray().add(array);
331         });
332     });
333     objectElement.getElementText().add(elementTextModule);
334 }

```

A.4 Gerar código-fonte React Native

O código-fonte do React Native é construído a partir da aplicação de regras de transformação no Modelo React-Native. A regra de transformação a seguir gera o código-fonte React Native do domínio modelado.

Código-fonte 5 – Regras de transformação em Java - Modelo React-Native para código-fonte React-Native

```

1 package reactnative.education.emf.implement;
2 import org.eclipse.emf.common.util.EList;
3 import reactNative.Array;
4 import reactNative.Button;
5 import reactNative.Component;
6 import reactNative.DataJS;
7 import reactNative.FlatList;
8 import reactNative.Function;

```

```

9 import reactNative.Icon;
10 import reactNative.Image;
11 import reactNative.ObjectElement;
12 import reactNative.ReactNativeFactory;
13 import reactNative.SectionList;
14 import reactNative.StyleSheet;
15 import reactNative.Text;
16 import reactNative.TouchableOpacity;
17 import reactNative.View;
18 public class ComponetsReactNative {
19     protected StringBuilder js;
20     protected String ql = "\n";
21     protected ReactNativeFactory reactNativeFactory = ReactNativeFactory.eINSTANCE
        ;
22     protected TouchableOpacity touchableOpacity;
23     private boolean isView;
24     protected StringBuilder flatToText;
25     private boolean isStyle = false;
26
27     public void setFlatToText(StringBuilder flatToText) {
28         this.flatToText = flatToText;
29     }
30
31     public StringBuilder getFlatToText() {
32         return flatToText;
33     }
34
35     private StringBuilder getComponentToText(Component component) {
36         js = new StringBuilder();
37         component.getComponentTotext().forEach(hasText -> {
38             js.append(getText(hasText));
39         });
40         return js;
41     }
42
43     protected StringBuilder ComponetToComponents(Component component) {
44         js = new StringBuilder();
45         js.append(this.getComponentToText(component));
46         js.append(this.getComponentToTouchableOpacity(component));
47         return js;
48     }
49
50     protected StringBuilder componentsToView(Component component) {
51         js = new StringBuilder();
52         component.getComponentToView().forEach(view -> {
53             js.append(this.getView(view));
54         });
55         return js;
56     }
57
58     protected StringBuilder componentToFunctions(Component component) {
59         StringBuilder functionBuilder = new StringBuilder();
60         component.getComponetToFunction().forEach(function -> {
61             functionBuilder.append(String.format("%s = (%s) =>{\n\t return(\n %s \n)}\n\n", function.getValue(),
62                 function.getParam() == null ? "" : function.getParam(), functionToView
                    (function)));

```



```

63
64     });
65     return functionBuilder;
66 }
67
68 private StringBuilder functionToView(Function function) {
69     StringBuilder functionToViewBuilder = new StringBuilder();
70     function.getFunctionToView().forEach(view -> {
71         functionToViewBuilder.append(this.getView(view));
72     });
73     return functionToViewBuilder;
74 }
75
76 private StringBuilder getViewToComponents(View view) {
77     StringBuilder viewToComponent = new StringBuilder();
78     view.getHasText().forEach(text -> {
79         viewToComponent.append("\t" + this.getText(text));
80     });
81     view.getHasImage().forEach(image -> {
82         viewToComponent.append("\t" + this.getImage(image));
83     });
84     view.getViewToIcon().forEach(icon -> {
85         viewToComponent.append("\t" + this.getIcon(icon));
86     });
87     view.getViewToTouchableOpacity().forEach(touchableOpacity -> {
88         viewToComponent.append("\t" + this.getTouchableOpacity(touchableOpacity));
89     });
90     view.getViewToButton().forEach(button -> {
91         viewToComponent.append("\t\t" + this.getButton(button));
92     });
93     view.getSubView().forEach(subView -> {
94         viewToComponent.append(this.getView(subView));
95     });
96     view.getViewToflatlists().forEach(flatlist -> {
97         viewToComponent.append(String.format("\t\t<FlatList data={%s} keyExtractor
= {%s} \nrenderItem={%s}/>\n",
98             flatlist.getData(), flatlist.getKeyExtractor(), flatlist.getRenderItem
());
99         viewToComponent.append(this.getFlatList(flatlist));
100     });
101
102     if (view.getViewToSectionList() != null) {
103         viewToComponent.append(String.format(
104             "\t\t<SectionList sections={%s} keyExtractor={%s} \
nrenderSelectionHeader={%s} \nrenderItem={%s}/>\n",
105             view.getViewToSectionList().getSections(), view.getViewToSectionList()
.getKeyExtractor(),
106             view.getViewToSectionList().getRenderSectionHeader(), view.
getViewToSectionList().getRenderItem());
107     }
108     return viewToComponent;
109 }
110
111 private StringBuilder getView(View hasView) {
112     StringBuilder view = new StringBuilder();
113     view.append(
114         String.format("\t<View %s>\n %s \t</View>\n", viewToStyleSheet(hasView),

```

```
        getViewToComponents(hasView)));
115     return view;
116 }
117
118 private StringBuilder viewToStyleSheet(View view) {
119     StringBuilder viewToStyle = new StringBuilder();
120     if (view.getViewToStyleSheet().isEmpty()) {
121         return viewToStyle.append("");
122     }
123     view.getViewToStyleSheet().forEach(styleSheet -> {
124         viewToStyle.append(this.styles(styleSheet));
125     });
126     return viewToStyle;
127 }
128
129 private StringBuilder textToStyleSheet(Text text) {
130     StringBuilder textToStyle = new StringBuilder();
131     if (text.getTextToStylesheet().isEmpty()) {
132         return textToStyle.append("");
133     }
134     text.getTextToStylesheet().forEach(styleSheet -> {
135         textToStyle.append(this.styles(styleSheet));
136     });
137     return textToStyle;
138 }
139
140 private StringBuilder imageToStyleSheet(Image image) {
141     StringBuilder imageToStyle = new StringBuilder();
142     if (image.getImageToStylesheet().isEmpty()) {
143         return imageToStyle.append("");
144     }
145     image.getImageToStylesheet().forEach(styleSheet -> {
146         imageToStyle.append(this.styles(styleSheet));
147     });
148     return imageToStyle;
149 }
150
151 private StringBuilder iconToStyleSheet(Icon icon) {
152     StringBuilder iconToStyle = new StringBuilder();
153     if (icon.getIconToStylesheet().isEmpty()) {
154         return iconToStyle.append("");
155     }
156     icon.getIconToStylesheet().forEach(styleSheet -> {
157         iconToStyle.append(this.styles(styleSheet));
158     });
159     return iconToStyle;
160 }
161
162 private StringBuilder styles(StyleSheet styleSheet) {
163     StringBuilder stringBuilder = new StringBuilder();
164     stringBuilder.append(String.format("style = {styles.%s}", styleSheet.
165         getNameStyle()));
166     return stringBuilder;
167 }
168 private StringBuilder getStyleSheetFunction(EList<Function> functions) {
169     StringBuilder sSheet = new StringBuilder("");
```

```

170     functions.forEach(function -> {
171         function.getFunctionToView().forEach(view -> {
172             sSheet.append(displayViewToStyleSheet(view));
173         });
174     });
175     return sSheet;
176 }
177
178 protected StringBuilder getStyleSheetComponent(Component component) {
179     StringBuilder sSheet = new StringBuilder();
180     component.getComponentToView().forEach(viewComponet -> {
181         if (viewComponet.getViewToStyleSheet().isEmpty()) {
182             return;
183         }
184         sSheet.append(
185             String.format("const styles = StyleSheet.create ({%s%s})",
displayViewToStyleSheet(viewComponet),
186                 getStyleSheetFunction(component.getComponentToFunction())));
187     });
188     return sSheet;
189 }
190
191 private StringBuilder displayViewToStyleSheet(View viewStyle) {
192     StringBuilder sSheet = new StringBuilder();
193     if (viewStyle.getViewToStyleSheet().isEmpty()) {
194         return sSheet.append("");
195     }
196     viewStyle.getViewToStyleSheet().forEach(style -> {
197         if (style.getNameStyle().isEmpty()) {
198             return;
199         }
200         if (!style.getNameStyle().isEmpty()) {
201             sSheet.append(
202                 String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(), this.
getStyleSheetAttr(style)));
203         }
204     });
205     viewStyle.getSubView().forEach(subView -> {
206         subView.getViewToStyleSheet().forEach(style -> {
207             if (!style.getNameStyle().isEmpty()) {
208                 sSheet.append(String.format("\n\t%s: {\n%s \t}, \t\n", style.
getNameStyle(),
209                     this.getStyleSheetAttr(style)));
210             }
211         });
212         subView.getViewToIcon().forEach(icon -> {
213             icon.getIconToStylesheet().forEach(style -> {
214                 if (style.getNameStyle().isEmpty()) {
215                     return;
216                 }
217                 sSheet.append(
218                     String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(),
this.getStyleSheetAttr(style)));
219             });
220         });
221         subView.getHasText().forEach(text -> {
222             text.getTextToStylesheet().forEach(style -> {

```

```
223     if (style.getNameStyle().isEmpty()) {
224         return;
225     }
226     sSheet.append(
227         String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(),
228 this.getStyleSheetAttr(style)));
229 });
230 subView.getHasImage().forEach(image -> {
231     image.getImageTostylesheet().forEach(style -> {
232         if (style.getNameStyle().isEmpty()) {
233             return;
234         }
235         sSheet.append(
236             String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(),
237 this.getStyleSheetAttr(style)));
238     });
239 });
240 viewStyle.getHasText().forEach(text -> {
241     text.getTextTostylesheet().forEach(style -> {
242         if (style.getNameStyle().isEmpty()) {
243             return;
244         }
245         sSheet.append(
246             String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(), this.
247 getStyleSheetAttr(style)));
248     });
249 });
250 viewStyle.getHasImage().forEach(image -> {
251     image.getImageTostylesheet().forEach(style -> {
252         if (style.getNameStyle().isEmpty()) {
253             return;
254         }
255         sSheet.append(
256             String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(), this.
257 getStyleSheetAttr(style)));
258     });
259 });
260 viewStyle.getViewToIcon().forEach(icon -> {
261     icon.getIconTostylesheet().forEach(style -> {
262         if (style.getNameStyle().isEmpty()) {
263             return;
264         }
265         sSheet.append(
266             String.format("\n\t%s: {\n%s \t}, \t\n", style.getNameStyle(), this.
267 getStyleSheetAttr(style)));
268     });
269 });
270 return sSheet;
271 }
272
273 private StringBuilder getStyleSheetAttr(StyleSheet styleSheet) {
274     StringBuilder attrStyle = new StringBuilder();
275     if (styleSheet.getHasFlexbox() == null) {
276         attrStyle.append("");
277     } else {
```

```
275     attrStyle.append(styleSheet.getHasFlexbox().getAlignItems() == null ? ""
276       : String.format("\t alignItems: %s, \n", styleSheet.getHasFlexbox().
getAlignItems(),
277         styleSheet.getHasFlexbox().getAlignItems()));
278     attrStyle.append(styleSheet.getHasFlexbox().getFlexDirection() == null ? "
"
279       : String.format("\t flexDirection: %s, \n", styleSheet.getHasFlexbox().
.getFlexDirection(),
280         styleSheet.getHasFlexbox().getFlexDirection()));
281     attrStyle.append(styleSheet.getHasFlexbox().getJustifyContent() == null ?
""
282       : String.format("\t justifyContent: '%s', \n", styleSheet.
getHasFlexbox().getJustifyContent(),
283         styleSheet.getHasFlexbox().getJustifyContent()));
284     attrStyle.append(styleSheet.getHasFlexbox().getFlex() == null ? ""
285       : String.format("\t flex: %s, \n", styleSheet.getHasFlexbox().getFlex
(),
286         styleSheet.getHasFlexbox().getFlex()));
287   }
288   if (styleSheet.getHasFont() != null) {
289     attrStyle.append(styleSheet.getHasFont().getFontFamily() == null ? ""
290       : String.format("\t fontFamily: %s, \n", styleSheet.getHasFont().
getFontFamily(),
291         styleSheet.getHasFont().getFontFamily()));
292     attrStyle.append(styleSheet.getHasFont().getFontSize() == null ? ""
293       : String.format("\t fontSize: %s, \n", styleSheet.getHasFont().
getFontSize(),
294         styleSheet.getHasFont().getFontSize()));
295     attrStyle.append(styleSheet.getHasFont().getFontWeight() == null ? ""
296       : String.format("\t fontWeight: %s, \n", styleSheet.getHasFont().
getFontWeight(),
297         styleSheet.getHasFont().getFontWeight()));
298   }
299
300   if (styleSheet.getHasMargin() != null) {
301     attrStyle.append(styleSheet.getHasMargin().getMarginTop() == null ? ""
302       : String.format("\t marginTop: %s, \n", styleSheet.getHasMargin().
getMarginTop(),
303         styleSheet.getHasMargin().getMarginTop()));
304     attrStyle.append(styleSheet.getHasMargin().getMarginBottom() == null ? ""
305       : String.format("\t marginBottom: %s, \n", styleSheet.getHasMargin().
getMarginBottom(),
306         styleSheet.getHasMargin().getMarginBottom()));
307     attrStyle.append(styleSheet.getHasMargin().getMargin() == null ? ""
308       : String.format("\t margin: %s, \n", styleSheet.getHasMargin().
getMargin(),
309         styleSheet.getHasMargin().getMargin()));
310   }
311
312   if (styleSheet.getHasBorder() != null) {
313     attrStyle.append(styleSheet.getHasBorder().getBorderBottomRightRadius() ==
null ? ""
314       : String.format("\t borderBottomRightRadius: %s, \n",
315         styleSheet.getHasBorder().getBorderBottomRightRadius(),
316         styleSheet.getHasBorder().getBorderBottomRightRadius()));
317     attrStyle.append(styleSheet.getHasBorder().getBorderColor() == null ? ""
318       : String.format("\t borderColor: %s, \n", styleSheet.getHasBorder().
```

```

getBorderColor(),
319         styleSheet.getHasBorder().getBorderColor());
320     attrStyle.append(styleSheet.getHasBorder().getBorderRadius() == null ? ""
321         : String.format("\t borderRadius: %s, \n", styleSheet.getHasBorder().
getBorderRadius(),
322         styleSheet.getHasBorder().getBorderRadius()));
323     attrStyle.append(styleSheet.getHasBorder().getBorderWidth() == null ? ""
324         : String.format("\t borderWidth: %s, \n", styleSheet.getHasBorder().
getBorderWidth(),
325         styleSheet.getHasBorder().getBorderWidth()));
326 }
327
328 if (styleSheet.getHasPadding() != null) {
329     attrStyle.append(styleSheet.getHasPadding().getPadding() == null ? ""
330         : String.format("\t padding: %s, \n", styleSheet.getHasPadding().
getPadding(),
331         styleSheet.getHasPadding().getPadding()));
332     attrStyle.append(styleSheet.getHasPadding().getPaddingHorizontal() == null
? ""
333         : String.format("\t paddingHorizontal: %s, \n", styleSheet.
getHasPadding().getPaddingHorizontal(),
334         styleSheet.getHasPadding().getPaddingHorizontal()));
335     attrStyle.append(styleSheet.getHasPadding().getPaddingVertical() == null ?
""
336         : String.format("\t paddingVertical: %s, \n", styleSheet.getHasPadding
().getPaddingVertical(),
337         styleSheet.getHasPadding().getPaddingHorizontal()));
338 }
339 return attrStyle;
340 }
341
342 protected StringBuilder getViewToFlatlist(Component component) {
343     this.flatToText = new StringBuilder();
344     component.getComponentToView().forEach(view -> {
345         view.getViewToflatlists().forEach(flatList -> {
346             this.flatToText.append(this.getFlatListToArray(flatList));
347         });
348         this.flatToText.append(this.getSectionListArray(view.getViewToSectionList
()));
349     });
350 }
351 return this.flatToText;
352 }
353
354 protected StringBuilder getViewToSectionList(Component component) {
355     StringBuilder sectionListBuilder = new StringBuilder();
356     component.getComponentToView().forEach(view -> {
357         sectionListBuilder.append(this.getSectionListArray(view.
getViewToSectionList()));
358     });
359 }
360 return sectionListBuilder;
361 }
362
363 protected StringBuilder getDataJS(DataJS dataJS) {
364     StringBuilder dataJSBuilder = new StringBuilder();
365     dataJS.getDataJsToArray().forEach(array -> {

```

```
366     dataJSBuilder.append(this.getArrayToObject(array));
367   });
368   return dataJSBuilder;
369 }
370
371 private StringBuilder getSectionListArray(SectionList sectionList) {
372   StringBuilder aSectionList = new StringBuilder();
373   if (sectionList != null) {
374     sectionList.getSectionListToArray().forEach(array -> {
375       aSectionList.append(String.format("%s:[%s]", array.getValue(), this.
376         getArrayToObject(array)));
377     });
378   }
379   return aSectionList;
380 }
381
382 private StringBuilder getFlatListToArray(FlatList flatList) {
383   StringBuilder aFlatlist = new StringBuilder();
384   flatList.getFlatListToArray().forEach(array -> {
385     aFlatlist.append(String.format("%s:[%s]", array.getValue(), this.
386       getArrayToObject(array)));
387   });
388   return aFlatlist;
389 }
390
391 private StringBuilder getArrayToObject(Array array) {
392   StringBuilder arrayBuilder = new StringBuilder();
393   array.getArrayToObjects().forEach(object -> {
394     arrayBuilder.append(String.format("\r\n\t{%s %s %s}", object.isId() ? "id
395       : Math.random(), " : ",
396       this.getObjectToElements(object), this.getObjectToArray(object)));
397   });
398   return arrayBuilder;
399 }
400
401 private StringBuilder getObjectToElements(ObjectElement object) {
402   StringBuilder arrayBuilder = new StringBuilder();
403   object.getElementText().forEach(text -> {
404     arrayBuilder.append(String.format("%s: '%s',\r\n\t", text.getValueText(),
405       text.getContentText()));
406   });
407   object.getElementTextToArray().forEach(array -> {
408     arrayBuilder.append(String.format(" %s: [%s],\r\n\t", array.getValue(),
409       getArrayToObject(array)));
410   });
411   object.getElementImage().forEach(image -> {
412     arrayBuilder.append(String.format("%s: require ('%s'), ", image.
413       getValueImage(), image.getContentImage()));
414     image.getElementImageToArray().forEach(array -> {
415       arrayBuilder.append(String.format(" %s: [%s],\r\n\t", array.getValue(),
416         getArrayToObject(array)));
417     });
418   });
419   return arrayBuilder;
420 }
```

```
416 private StringBuilder getObjectToArray(ObjectElement object) {
417     StringBuilder arrayBuilder = new StringBuilder();
418     object.getObjectToArray().forEach(array -> {
419         arrayBuilder.append(String.format("%s: [%s]", array.getValue(), this.
420             getObjectToArray(array)));
421     });
422     return arrayBuilder;
423 }
424 private StringBuilder getFlatList(FlatList hasFlatList) {
425     StringBuilder flatlist = new StringBuilder();
426     hasFlatList.getHasViews().forEach(view -> {
427         flatlist.append(this.getView(view));
428     });
429     return flatlist;
430 }
431 private StringBuilder getText(Text hasText) {
432     StringBuilder text = new StringBuilder();
433     text.append(String.format("\t<Text %s> %s </Text> %s", this.textToStyleSheet
434         (hasText), hasText.getContentText(),
435         ql));
436     return text;
437 }
438 private StringBuilder getImage(Image hasImage) {
439     StringBuilder image = new StringBuilder();
440     image.append(
441         String.format("\t<Image %s source={%s}/> %s", imageToStyleSheet(hasImage
442             ), hasImage.getSource(), ql));
443     return image;
444 }
445 private StringBuilder getIcon(Icon hasIcon) {
446     StringBuilder icon = new StringBuilder();
447     icon.append(String.format("\t<Icon name='%s' size={%d} color='%s' %s/> \n",
448         hasIcon.getName(),
449         hasIcon.getSize(), hasIcon.getColor(), iconToStyleSheet(hasIcon)));
450     return icon;
451 }
452 private StringBuilder getButton(Button hasButton) {
453     StringBuilder button = new StringBuilder();
454     if (hasButton.getOnPress() == null) {
455         hasButton.setOnPress("");
456     } else if (hasButton.getTitle() == null) {
457         hasButton.setTitle("");
458     }
459     button.append(
460         (hasButton.getOnPress().isEmpty()) ? String.format("\t<Button title={%s}
461             />\n", hasButton.getTitle())
462         : String.format("\t<Button title={%s} onPress={%s} />\n", hasButton.
463             getTitle(),
464             hasButton.getOnPress()));
465     return button;
466 }
```



```

467 private StringBuilder getTouchableOpacity(TouchableOpacity hasTouchableOpacity
468 ) {
469     StringBuilder touchableOpacity = new StringBuilder();
470     this.isView = true;
471     touchableOpacity.append(isOnPress(hasTouchableOpacity, isView));
472     return touchableOpacity;
473 }
474 private StringBuilder getComponentToTouchableOpacity(Component component) {
475     js = new StringBuilder();
476     isView = false;
477     if (component.getComponentToTouchableOpacity() == null) {
478         return js;
479     }
480     js.append(isOnPress(component.getComponentToTouchableOpacity(), isView));
481     return js;
482 }
483
484 private StringBuilder isOnPress(TouchableOpacity touchableOpacity, boolean
485 isView) {
486     StringBuilder js = new StringBuilder();
487     if (touchableOpacity.getOnPress() == null) {
488         touchableOpacity.setOnPress("");
489     }
490     if (isView == false) {
491         if (touchableOpacity.getOnPress().isEmpty()) {
492             js.append(String.format("\t<TouchableOpacity>\n %s \t</TouchableOpacity>
493 ",
494                 getTouchableOpacityToComponents(touchableOpacity)));
495         } else {
496             js.append(String.format("\t<TouchableOpacity onPress={%s}>\n %s \t</
497 TouchableOpacity>",
498                 touchableOpacity.getOnPress(), getTouchableOpacityToComponents(
499 touchableOpacity)));
500         }
501     } else {
502         if (touchableOpacity.getOnPress().isEmpty()) {
503             js.append(String.format("\t<TouchableOpacity>\n %s \t\t</
504 TouchableOpacity> %s",
505                 getTouchableOpacityToComponents(touchableOpacity), ql));
506         } else {
507             js.append(String.format("\t<TouchableOpacity onPress={%s}>\n %s \t\t</
508 TouchableOpacity> %s",
509                 touchableOpacity.getOnPress(), getTouchableOpacityToComponents(
510 touchableOpacity), ql));
511         }
512     }
513     return js;
514 }
515 private StringBuilder getTouchableOpacityToComponents(TouchableOpacity
516 touchableOpacity) {
517     StringBuilder js = new StringBuilder();
518     touchableOpacity.getTouchableOpacityToIcon().forEach(icon -> {
519         js.append("\t" + this.getIcon(icon));
520     });

```

```
515     touchableOpacity.getTouchableOpacityToText().forEach(text -> {
516         js.append("\t" + this.getText(text));
517     });
518     touchableOpacity.getTouchableOpacityToImage().forEach(image -> {
519         js.append("\t" + this.getImage(image));
520     });
521     touchableOpacity.getTouchableOpacityView().forEach(view -> {
522         js.append(this.getView(view));
523     });
524     touchableOpacity.getTouchableOpacityToButton().forEach(button -> {
525         js.append("\t" + this.getButton(button));
526     });
527     return js;
528 }
529
530 }
```

INSTRUMENTAÇÃO DA AVALIAÇÃO

B.1 Questionário de avaliação da Educom.ml

Avaliação da Abordagem de Modelagem Educom.ml (Education content modeling for M-Learning)

Este questionário tem como objetivo avaliar a Educom.ml, segundo a visão dos professores que participaram do experimento.

***Obrigatório**

1. E-mail *

2. Eu considero útil a estruturação do conteúdo educacional proposto pela Educom.ml.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

Observações.

3. Eu acredito que a abordagem de modelagem Educom.ml ajuda no processo de produção do conteúdo educacional.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

Observações.

4. Eu considero que os conceitos apresentados no Modelo Conceitual, proposto pela Educom.ml, são claros e de fácil compreensão.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

5. Eu considero que os conceitos apresentados no Modelo Instrucional, proposto pela Educom.ml, são claros e de fácil compreensão.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

6. Eu considero que os conceitos apresentados no Modelo Didático, propostos pela Educom.ml, são claros e de fácil compreensão.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

7. Eu considero que as diretrizes de produção de mídia (texto, imagem, vídeo) apresentados pela Educom.ml são claras e de fácil compreensão.

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

8. Após conhecer a Educom.ml, você a adotaria para produzir conteúdo educacional para aprendizagem móvel?

Marcar apenas uma oval.

- Discordo fortemente
- Discordo
- Não concordo, nem discordo
- Concordo
- Concordo fortemente

9. Utilize este espaço para deixar sugestões de melhoria da modelagem de conteúdo educacional apresentada.

B.2 Questionário de perfil do participante

Perfil do Participante

Este formulário tem o objetivo de colher dados que caracterizam os participantes do estudo. O acesso e a análise dos dados coletados serão realizados apenas pelo pesquisador e seu orientador.

***Obrigatório**

1. E-mail *

2. Qual sua idade?

3. Qual sua formação acadêmica? *

Marque todas que se aplicam.

- Graduação
 Especialização
 Mestrado
 Doutorado
 Pós-Doutorado

4. 3. Qual a categoria administrativa de sua instituição de ensino?

Marque todas que se aplicam.

- Pública
 Privada

5. Com qual nível de ensino você trabalha/trabalhou?

Marque todas que se aplicam.

- Ensino infantil
- Ensino fundamental
- Ensino médio
- Graduação
- Pós-graduação
- Cursos profissionalizantes

6. Indique com quais modalidades de ensino e aprendizagem você trabalha/trabalhou?

Marque todas que se aplicam.

- Presencial
- Ensino a distância

7. Indique qual o seu tempo de experiência no ensino presencial?

Marcar apenas uma oval.

- Até 1 ano
- De 2 a 4 anos
- De 5 a 10 anos
- Acima de 10 anos
- não tenho

8. Indique qual o seu tempo de experiência no ensino a distância?

Marcar apenas uma oval.

- Até 1 ano
- De 2 a 4 anos
- De 5 a 10 anos
- Acida de 10 anos
- não tenho

9. Você trabalha ou trabalhou com alguma ferramenta de modelagem de dados, de software ou de processos (por exemplo: UML - Unified Modeling Language, DER - Diagramas Entidade-Relacionamento, BPMN - Business Process Model and Notation)?

Marcar apenas uma oval.

Sim

Não

10. No caso afirmativo, indique qual a ferramenta de modelagem utilizou?

11. 9. Você já produziu conteúdo didático para alguma das modalidades de ensino e aprendizagem a seguir?

Marque todas que se aplicam.

e-learning - modalidade de ensino e aprendizagem que explora as TIC's para facilitar o acesso a evolução, a melhoria da qualidade da educação e formação (Vlachopoulos, Cabrera & Bravo (2011))

m-learning - modalidade de ensino e aprendizagem em que dispositivos móveis são utilizados como mediador no processo de ensino e aprendizagem (Alenxandre - 2004)

t-learning - modalidade de ensino e aprendizagem que utiliza a tecnologia de TV Digital Interativa no processo de formação educacional (VÉRAS, Douglas et al (2009)).

u-learning - aprendizagem ubíqua pode ser definida como a utilização de dispositivos e tecnologias móveis, sensores e mecanismos de localização, os quais levam em consideração características particulares dos estudantes, objetivando auxiliar no processo educacional (Parise, Douglas, et al (2014)).

clássico - modalidade de ensino presencial.

Outro: _____

12. Você costuma usar alguma metodologia para produzir conteúdo didático de cursos que ensina?

Marcar apenas uma oval.

Sim

Não

13. No caso afirmativo, indique qual metodologia de produção de material didático utiliza.

14. Eu produzo material didático que são fáceis de serem adaptados a objetivos diferentes em um tempo razoável.

Marcar apenas uma oval.

- discordo totalmente
- discordo
- não concordo, nem concordo
- concordo
- concordo totalmente

15. Eu tenho facilidade em utilizar tecnologias.

Marcar apenas uma oval.

- discordo totalmente
- discordo
- não concordo, nem concordo
- concordo
- concordo totalmente

16. Eu frequentemente utilizo tecnologias nos cursos que ensino (vídeos, podcasts, sites, plataformas EAD, entre outros).

Marcar apenas uma oval.

- nunca
- raramente
- às vezes
- frequentemente
- sempre

17. Você considera interessante ter uma abordagem de modelagem que guia o professor na elaboração de conteúdo didático para aplicativos móveis?

Marcar apenas uma oval.

- discordo totalmente
- discordo
- não concordo, nem discordo
- concordo
- concordo totalmente

