



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO**

EDGAR MARÇAL DE BARROS FILHO

**UFC-INVENTOR: UM AMBIENTE PARA MODELAGEM E GERAÇÃO DE
APLICAÇÕES PARA AULAS DE CAMPO UBÍQUAS**

FORTALEZA

2016

EDGAR MARÇAL DE BARROS FILHO

UFC-INVENTOR: UM AMBIENTE PARA MODELAGEM E GERAÇÃO DE
APLICAÇÕES PARA AULAS DE CAMPO UBÍQUAS

Tese de Doutorado apresentada ao Programa de
Mestrado e Doutorado em Computação da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de Doutor em
Ciência da Computação.

Orientador: Prof. Dra. Rossana Maria de
Castro Andrade.

Coorientador: Prof. Dr. Windson Viana de
Carvalho.

FORTALEZA

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

F498u Filho, Edgar Marçal de Barros.

UFC-INVENTOR: UM AMBIENTE PARA MODELAGEM E GERAÇÃO DE APLICAÇÕES PARA AULAS DE CAMPO UBÍQUAS / Edgar Marçal de Barros Filho. – 2016.
165 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2016.

Orientação: Profa. Dra. Rossana Maria de Castro Andrade.

Coorientação: Prof. Dr. Windson Viana de Carvalho.

1. Ubiquitous learning. 2. Desenvolvimento dirigido por modelos. 3. Aulas de campo ubíquas. 4. Aplicações multiplataformas. I. Título.

CDD 005

EDGAR MARÇAL DE BARROS FILHO

UFC-INVENTOR: UM AMBIENTE PARA MODELAGEM E GERAÇÃO DE
APLICAÇÕES PARA AULAS DE CAMPO UBÍQUAS

Tese de Doutorado apresentada ao Programa de
Mestrado e Doutorado em Computação da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de Doutor em
Ciência da Computação.

Aprovada em ____/____/____ .

BANCA EXAMINADORA

Prof.^a. Dr.^a. Rossana Maria de Castro Andrade (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Windson Viana de Carvalho (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Armando Valente
Universidade Estadual de Campinas (Unicamp)

Prof. Dr. José Valdeni de Lima
Universidade Federal do Rio Grande do Sul (UFRGS)

Prof. Dr. José Aires de Castro Filho
Universidade Federal do Ceará (UFC)

Prof. Dr. Francisco Heron de Carvalho Júnior
Universidade Federal do Ceará (UFC)

RESUMO

As aulas de campo promovem o espírito científico dos alunos por meio do aumento da capacidade de observação e descoberta em diversas áreas do conhecimento. Ao incorporar tecnologias ubíquas, como sensores e atuadores, nas aulas de campo (aqui denominadas Aulas de Campo Ubíquas), estas apresentam benefícios quando comparadas às tradicionais. Entretanto, o desenvolvimento de sistemas ubíquos impõe desafios, tais como: tratamento de forma transparente da heterogeneidade dos dispositivos; ausência de infraestrutura de comunicação fixa; dinamicidade dos elementos computacionais do ambiente; e o suporte à mobilidade dos usuários. Em levantamento bibliográfico realizado, não foi encontrada uma ferramenta que possibilitasse a modelagem e a geração de aplicações multiplataforma para aulas de campo ubíquas voltadas aos profissionais de ensino com pouco ou nenhum conhecimento sobre programação. Sendo assim, esta tese propõe um ambiente computacional, composto por uma linguagem de modelagem e dois sistemas integrados, para favorecer a utilização das tecnologias ubíquas em aulas de campo: o Ubiquitous Field Classes Inventor (UFC-Inventor). Com este ambiente, profissionais de ensino podem criar modelos gráficos de suas aulas de campo e, a partir deles, gerar aplicações ubíquas multiplataforma para serem usadas pelos alunos em campo. A construção do ambiente foi iniciada a partir da realização de um mapeamento sistemático com artigos que relatavam experiências sobre o uso das tecnologias ubíquas na educação. De acordo com esse levantamento, foi identificado um conjunto de requisitos essenciais para as aulas de campo ubíquas. Considerando esses requisitos, esta tese apresenta o UFC-Inventor e seus componentes: a ML4UL (*Modeling Language for Ubiquitous learning*), uma linguagem específica de domínio, cuja especificação é aberta e reutilizável, voltada para representação das aulas de campo ubíquas; o módulo UFC-GLM, para modelagem gráfica das aulas de campo com recursos de computação ubíqua; e, o módulo UFC-Generator, que possibilita a geração das aplicações ubíquas para diferentes sistemas operacionais. Para validar o ambiente, foi realizada uma avaliação com cinco professores sobre a usabilidade do ambiente e um estudo de caso em uma aula de campo com dezessete alunos. Considerando a análise das respostas aos questionários aplicados com os usuários e o funcionamento do ambiente no estudo de caso, os resultados sugerem a aceitação do ambiente e indicam a execução adequada do UFC-Inventor.

Palavras-chave: *Ubiquitous learning*; Desenvolvimento dirigido por modelos; Aulas de campo ubíquas; Aplicações multiplataformas.

ABSTRACT

Field trips (or field classes) promote the scientific spirit of the students by increasing the ability of observation and discovery in various areas of knowledge. By incorporating ubiquitous technologies, such as sensors and actuators, in the field classes (here called Ubiquitous Field Classes), they show benefits when compared to traditional classes. However, the development of ubiquitous systems has challenges, such as seamless treatment of the devices heterogeneity; absence of fixed communication infrastructure; dynamics of the computational elements of the environment; and support for user mobility. In the literature review, we have not found a tool that enables modeling and generation of cross-platform applications for ubiquitous field classes aimed at teaching professionals with little or no knowledge of programming. This document presents a doctoral thesis that consists of a computational environment called the Ubiquitous Field Classes Inventor (UFC-Inventor), which comprises of a modeling language and integrated systems. Together, they enhance the use of ubiquitous technologies in field classes. Thus, teaching professionals can create graphical models of their field classes and, from it, generate cross-platform ubiquitous applications to be used by students. First, we conducted a systematic mapping study with articles on experiences on the use of ubiquitous technology in education. From this survey, we identified some important requirements for ubiquitous field classes. From them, we designed and implemented the UFC-Inventor and its components. The first component is the ML4UL (Modeling Language for Ubiquitous Learning), a domain specific language focused on representation of ubiquitous field classes through an open and reusable specification. The UFC-GLM module, indeed, allows graphic modeling of the field classes with ubiquitous computing resources. The third module is the UFC-Generator, which is responsible for generating the ubiquitous applications. We carried out an evaluation of UFC Inventor with five teachers aim at verifying the usability of the environment. Also, we implemented a case study in the field with seventeen students. The results of both evaluation experiments suggest the acceptance of the UFC-Inventor and indicate that its execution has occurred correctly.

Keywords: Ubiquitous learning; Model-Driven Development; Ubiquitous Field Classes.

LISTA DE FIGURAS

FIGURA 1 – VISÃO GERAL DA METODOLOGIA.....	18
FIGURA 2 – EXEMPLOS DE AULAS DE CAMPO	22
FIGURA 3 – VISÃO DO SISTEMA GREAT TOUR	28
FIGURA 4 – VISÃO GERAL DA ENGENHARIA DIRIGIDA POR MODELOS	34
FIGURA 5 – EXEMPLO DE REPRESENTAÇÃO DA METAMODELAGEM	37
FIGURA 6 – REPRESENTAÇÃO DA TRANSFORMAÇÃO ENTRE MODELOS	38
FIGURA 7 – FASES DO MOBILE LEARNING, SEGUNDO SHARPLES E ROSCHELLE (2010).....	41
FIGURA 8 – ARQUITETURA TÍPICA DE AMBIENTES DE AUTORIA EM MOBILE LEARNING.	45
FIGURA 9 – EXEMPLO COM OS ELEMENTOS DA IMS LD EM UMA SALA DE AULA	48
FIGURA 10 – VISÃO GERAL DOS ELEMENTOS DO IMS LD	50
FIGURA 11 – ESTRUTURA DO IMS CONTENT PACKING	51
FIGURA 12 – DIAGRAMA DE ATIVIDADES COM EXEMPLO DE AULA PRESENCIAL.....	52
FIGURA 13 – EXEMPLO DE CODIFICAÇÃO EM IMS LD DO ELEMENTO <ROLE>	52
FIGURA 14 – EXEMPLO DE CODIFICAÇÃO EM IMS LD DO ELEMENTO <LEARNING-ACTIVITY>.....	53
FIGURA 15 – IMAGEM DO EDITOR IMS RELOAD.....	54
FIGURA 16 – A) TELA DA FERRAMENTA COMPENDIUM LD. B) TELA DA FERRAMENTA CADMOS LD	54
FIGURA 17 – A) TELA DA FERRAMENTA RELOAD PLAYER B) PÁGINA WEB COM A EXECUÇÃO DE UMA UOL EM IMS LD VIA COPPERCORE	55
FIGURA 18 – TELA DO AMBIENTE DE EDIÇÃO DO LEMONADE.....	59
FIGURA 19 – VISÃO GERAL DO MODELO PROPOSTO EM GÓMEZ (2013)	60
FIGURA 20 – TELA DO SISTEMA ARLEARN	62
FIGURA 21 – TELAS DO SISTEMA MAT FOR ARLEARN	62
FIGURA 22 – VISÃO GERAL DO AMBIENTE MIDGAR.....	64
FIGURA 23 – TELA DA FERRAMENTA APP INVENTOR	66
FIGURA 24 – PERFIL DAS AULAS DE CAMPO UBÍQUAS	73
FIGURA 25 – A) COMPARATIVO ENTRE OS SENSORES E AS INFORMAÇÕES CONTEXTUAIS; B)COMPARATIVO ENTRE OS SISTEMAS OPERACIONAIS E A ABORDAGEM DE DESENVOLVIMENTO.....	75
FIGURA 26 – VISÃO GERAL DO AMBIENTE UFC-INVENTOR.....	82
FIGURA 27 – TRANSFORMAÇÕES ENTRE MODELOS NO UFC-INVENTOR	83
FIGURA 28 – ILUSTRAÇÃO DE UMA AULA DE CAMPO COM SMARTPHONES E ELEMENTOS IMS LD.....	86
FIGURA 29 – VISÃO GERAL DOS ELEMENTOS PROPOSTOS NA ML4UL (EM AMARELO).....	87
FIGURA 30 – EXEMPLO EM DIAGRAMA DE ATIVIDADES E CODIFICAÇÃO EM IMS LD COM ML4UL	93
FIGURA 31 – VISÃO GERAL DO FUNCIONAMENTO DO MÓDULO UFC-GLM	95
FIGURA 32 – TELA DO MÓDULO UFC-GLM	96
FIGURA 33 – TELA DO UFC-GLM PARA CRIAÇÃO/EDIÇÃO DE UMA ATIVIDADE.....	97
FIGURA 34 – TELA DO UFC-GLM PARA CRIAÇÃO DAS PERGUNTAS DE UMA ATIVIDADE.	99
FIGURA 35 – TELA DO UFC-GLM PARA CONFIGURAÇÃO DAS REGRAS DE ADAPTAÇÃO.	100
FIGURA 36 – VISÃO GERAL DO PROCESSO DE TRANSCODIFICAÇÃO.	102

FIGURA 37 – TRECHO DE CÓDIGO DO ARQUIVO IMSMANIFEST.XML.....	103
FIGURA 38 – TRECHO DE CÓDIGO DO ARQUIVO MEDIA-CAPTURE.XSL	104
FIGURA 39 – TRECHO DE CÓDIGO DO ARQUIVO INDEX.HTML	104
FIGURA 40 – TELA DO SISTEMA UFC-GENERATOR.	105
FIGURA 41 – PROCESSO PARA AVALIAÇÃO DA USABILIDADE DO UFC-INVENTOR.....	110
FIGURA 42 – FREQUÊNCIA (%) DAS RESPOSTAS DA SEGUNDA PARTE DO INSTRUMENTO DE AVALIAÇÃO.	112
FIGURA 43 – PROCEDIMENTO REALIZADO PARA AVALIAÇÃO DO AMBIENTE UFC-INVENTOR	114
FIGURA 44 – MODELO DA AULA DE CAMPO PROJETADO NO MÓDULO UFC-GLM.	115
FIGURA 45 – IMAGENS DE VERSÕES DA APLICAÇÃO EXECUTADAS EM DIFERENTES DISPOSITIVOS MÓVEIS.	116
FIGURA 46 – USO DO APLICATIVO PELOS ALUNOS EM DIFERENTES MOMENTOS DA AULA DE CAMPO.	117
FIGURA 47 – ILUSTRAÇÃO DAS TRANSFORMAÇÕES ENTRE OS MODELOS DO ESTUDO DE CASO.	119
FIGURA 48 – MÉDIAS INDIVIDUAIS DOS ALUNOS NAS CARACTERÍSTICAS AVALIADAS.	122
FIGURA 49 – GRÁFICO BLOXSPOT RELACIONADO À PRIMEIRA PARTE DO INSTRUMENTO DE AVALIAÇÃO.	123
FIGURA 50 – FREQUÊNCIA (%) DAS RESPOSTAS DA SEGUNDA PARTE DO INSTRUMENTO DE AVALIAÇÃO.	125
FIGURA 51 – EXEMPLOS DE RESPOSTAS FOTOGRAFADAS ÀS QUESTÕES DA APLICAÇÃO.	126
FIGURA 52 – PÁGINA WEB PARA EXIBIÇÃO DOS DADOS DO ESTUDO DE CAMPO.	129

LISTA DE TABELAS

TABELA 1 – VANTAGENS E DESVANTAGENS DO USO DE DSL.....	35
TABELA 2 – EXEMPLOS DE SINTAXES ABSTRATA, CONCRETA E SEMÂNTICA DA LINGUAGEM SQL.....	36
TABELA 3 – COMPARAÇÃO ENTRE OS TRABALHOS RELACIONADOS	68
TABELA 4 – QUANTIDADE DE ARTIGOS RESULTANTES DO MAPEAMENTO SISTEMÁTICO	72
TABELA 5 – TAREFAS IDENTIFICADAS NO MAPEAMENTO AGRUPADAS POR CATEGORIAS	74
TABELA 6 – REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DE AULAS DE CAMPO UBÍQUAS	76
TABELA 7 – PERFIL DA APLICAÇÃO GEOMÓVEL.....	79
TABELA 8 – PERFIL DA APLICAÇÃO CAULS	80
TABELA 9 – TABELA 'REGISTROS' PARA ARMAZENAMENTO DAS INFORMAÇÕES EM SERVIDOR NA INTERNET	91
TABELA 10 – EXEMPLO DE PREENCHIMENTO DA TABELA 'REGISTROS' COM DADOS FICTÍCIOS	92
TABELA 11 – CLASSIFICAÇÃO DA USABILIDADE DO APLICATIVO CONFORME DUAS ESCALAS DIFERENTES.....	111
TABELA 12 – RESUMO DA ANÁLISE DA PRIMEIRA PARTE DO QUESTIONÁRIO DE AVALIAÇÃO.	121
TABELA 13 – CLASSIFICAÇÃO DA USABILIDADE DO APLICATIVO CONFORME DUAS ESCALAS DIFERENTES.....	121
TABELA 14 – RESUMO DOS DADOS REFERENTES AOS RESULTADOS DA PRIMEIRA PARTE DA AVALIAÇÃO.	123
TABELA 15 – RESULTADO DA AVALIAÇÃO DA UTILIDADE PERCEBIDA DA APLICAÇÃO	124
TABELA 16 – FORMA ESCOLHIDA PARA RESPONDER AS PERGUNTAS	126
TABELA 17 – COMPARAÇÃO ENTRE OS TRABALHOS RELACIONADOS E O UFC-INVENTOR.	135
TABELA 18 – ARTIGOS CIENTÍFICOS PRODUZIDOS	136

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ATL	<i>Atlas Transformation Language</i>
CSS	<i>Cascading Style Sheets</i>
DM	Dispositivo Móvel
DSL	<i>Domain Specific Language</i>
EML	<i>Educational Modeling Language</i>
GPS	<i>Global Positioning System</i>
LD	<i>Learning Design</i>
LOM	<i>Learning Object Metadata</i>
LPS	Linha de Produto de Software
M2M	<i>Model-to-Model</i>
M2T	<i>Model-to-Text</i>
MAT	<i>Mobile Authoring Tool</i>
MDA	<i>Model-Driven Architecture</i>
MDD	<i>Model-Driven Development</i>
MDE	<i>Model-Driven Engineering</i>
MOF	<i>Meta Object Facility</i>
OA	Objeto de Aprendizagem
OCL	<i>Object Constraint Language</i>
OSGi	<i>Open Services Gateways initiative</i>
PDA	<i>Personal Digital Assistants</i>
QP	Questão de pesquisa
RAFT	<i>Remote Accessible Field Trips</i>
SCORM	<i>Shareable Content Object Reference Model</i>
SO	Sistema Operacional
T2M	<i>Text-to-Model</i>
UFC	<i>Ubiquitous Field Class</i>
UFC	Universidade Federal do Ceará
UML	<i>Unified Modeling Language</i>
UoL	<i>Unit of Learning</i>
W3C	<i>World Wide Web Consortium</i>

XML	<i>eXtensible Markup Language</i>
XSLT	<i>eXtensible Stylesheet Language for Transformation</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Hipótese e questões de pesquisa.....	16
1.3	Objetivo principal e metas	17
1.4	Metodologia	18
1.5	Organização do documento	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Aulas de Campo	21
2.3	Desenvolvimento dirigido por modelos.....	33
2.4	<i>Mobile Learning e Ubiquitous Learning</i>	40
2.5	Modelagem de processos e recursos educacionais	45
2.6	Conclusão.....	57
3	TRABALHOS RELACIONADOS	58
3.1	Lemonade (GIEMZA; BOLLEN; HOPPE, 2011)	58
3.2	UoLmP (GÓMEZ, 2013)	59
3.3	MAT for ARLearn (TABUENCA <i>et al.</i>, 2014).....	61
3.4	Midgar (GARCÍA <i>et al.</i>, 2014).....	63
3.5	App Inventor (MIT, 2012).....	65
3.6	Análise comparativa	66
3.7	Conclusão	70
4	REQUISITOS DE SISTEMAS UBÍQUOS PARA AULAS DE CAMPO	71
4.1	Mapeamento sistemático	71
4.2	Definição dos requisitos	76
4.3	Exemplos de implementações dos requisitos.....	78
4.4	Conclusão	80
5.	O AMBIENTE UFC-INVENTOR	81
5.1	Visão geral	81
5.2	A linguagem ML4UL.....	85
5.3	O módulo UFC-GLM	94
5.4	O módulo UFC-Generator	101
5.5	Conclusão	105
6.	AVALIAÇÃO DO AMBIENTE UFC-INVENTOR.....	106

6.1	Metodologia.....	106
6.2	Instrumentos	107
6.3	Análise da usabilidade do UFC-Inventor	109
6.4	Estudo de caso	112
6.5.	Conclusão	132
7.	CONCLUSÕES E TRABALHOS FUTUROS.....	133
7.1.	Visão geral do trabalho	133
7.2.	Principais contribuições e resultados alcançados	133
7.3.	Análise da hipótese e das questões de pesquisa.....	137
7.4.	Limitações da Tese.....	139
7.5.	Trabalhos futuros.....	140
	REFERÊNCIAS	143
	APÊNDICE A – Listagem das Referências dos 63 Artigos Analisados no Mapeamento Sistemático	156
	APÊNDICE B – Instrumentos de Avaliação	161
	APÊNDICE C – Respostas dos Alunos à Primeira Parte do Questionário de Avaliação	164

1 INTRODUÇÃO

Este capítulo introduz a tese de doutorado cujo objetivo é propor o UFC-Inventor, um ambiente para modelagem e geração de aplicações ubíquas com foco em aulas de campo. Na Seção 1.1 deste capítulo são realizadas a contextualização e a caracterização do problema abordado neste trabalho. A hipótese e as questões de pesquisa que nortearam o desenvolvimento deste trabalho são apresentadas na Seção 1.2. Na Seção 1.3, os objetivos e as principais metas são listados. Em seguida, na Seção 1.4, a metodologia utilizada para a elaboração deste trabalho é descrita. Por fim, a Seção 1.5 encerra o capítulo apresentando a organização do restante deste documento.

1.1 Contextualização

A aula de campo é uma metodologia que proporciona diferentes benefícios à aprendizagem, caracterizando-se como uma ação motivadora que permite aos alunos vivenciarem na prática o que foi lecionado em sala de aula (SHAKIL; FAIZI; HAFEEZ, 2011). Ela favorece uma abordagem ao mesmo tempo mais complexa e menos abstrata dos fenômenos estudados em diferentes domínios do conhecimento, ampliando a capacidade de observação e descoberta dos alunos. As atividades desenvolvidas durante as aulas de campo são essenciais para a aprendizagem em diversos cursos de graduação (e.g. geologia, geografia e engenharia de minas) e importantes práticas educativas no ensino básico, em assuntos como ecologia e meio ambiente.

Entretanto, durante as aulas de campo, alunos e professores se deparam com obstáculos que podem comprometer os benefícios dessas práticas educativas (ROSLIN *et al.*, 2013; SHAKIL; FAIZI; HAFEEZ, 2011). Em determinadas situações, os alunos ficam sobrecarregados com o excesso de atividades (e.g., analisar e registrar as informações do ambiente, anotar as observações do professor e realizar as atividades em grupo), além de terem de carregar diversos equipamentos (e.g., GPS, bússola, câmera digital e caderneta de campo). Já os professores muitas vezes têm de lidar com turmas numerosas e alunos dispersos no campo, o que pode resultar em um suporte inadequado à aprendizagem dos estudantes.

As tecnologias ubíquas têm sido utilizadas para auxiliar os alunos durante as atividades de aprendizagem nas aulas de campo (CHIANG; YANG; HWANG, 2014; CHIN; CHEN, 2013; HWANG; TSAI, YANG, 2008). Nesse trabalho, utilizaremos o termo Aulas de

Campo Ubíquas (UFC – *Ubiquitous Field Classes*) para referenciar as práticas educativas realizadas em campo com suporte das tecnologias ubíquas.

A integração dos recursos da computação ubíqua nessas práticas educativas pode proporcionar novas possibilidades aos processos de ensino e aprendizagem (GIEMZA; BOLLEN; HOPPE, 2011; MARÇAL *et al.*, 2013; WU; HWANG; TSAI, 2013), dentre as quais, podemos citar: identificação do contexto do aluno para entrega de conteúdos e exercícios de acordo com a situação dele naquele momento; registro das ações e preferências dos aprendizes para recomendações futuras em atividades similares; utilização de sensores (embutidos ou acoplados ao dispositivo móvel) para coleta e anotação de informações relevantes à aprendizagem em campo; e, comunicação entre alunos e professores através de redes sem fio para esclarecimento de dúvidas ou compartilhamento de ideias.

Entretanto, o desenvolvimento de sistemas ubíquos impõe desafios, tais como: tratamento de forma transparente da heterogeneidade dos dispositivos; ausência de infraestrutura de comunicação fixa; dinamicidade dos elementos computacionais do ambiente; e o suporte à mobilidade dos usuários (CASSOU *et al.*, 2009; KINDBERG; FOX, 2002; MAIA; ROCHA; ANDRADE, 2009). Além disso, a reutilização de sistemas existentes com fins educativos similares não é um processo fácil, devido principalmente à dificuldade ou à impossibilidade de se realizar customizações necessárias (e.g. inclusão de novos conteúdos e exercícios ou mudança de idioma). Dessa forma, o uso das tecnologias ubíquas em aulas de campo não é uma tarefa trivial para professores e outros profissionais de ensino sem conhecimentos de programação.

Uma abordagem utilizada para a construção de sistemas ubíquos que tem entre seus benefícios a abstração dos aspectos complexos da programação é a Engenharia Dirigida por Modelos (em inglês, *Model-Driven Engineering* - MDE) (BRAMBILLA; CABOT; WIMMER, 2012). Através das ferramentas baseadas na MDE, é possível a representação dos conceitos do domínio de forma efetiva e a geração automática dos sistemas, através de transformações entre modelos. Sistemas de diferentes áreas do conhecimento têm sido construídos utilizando-se as tecnologias de desenvolvimento dirigido por modelos (CUADRADO; CÁNOVAS IZQUIERDO; MOLINA, 2014, HERMIDA *et al.*, 2013).

Na área educacional, destacam-se as ferramentas de autoria que facilitam a construção de aplicações voltadas para apoiar as práticas educativas. Giemza, Bollen e Hoppe (2011) apresentam um software que possibilita o projeto de aulas de campo e a exportação de aplicações para os alunos utilizarem em dispositivos móveis com sistema operacional (SO) Windows Phone. Outra ferramenta é a MAT for ARLearn (Mobile Authoring Tool for

ARLearn), que permite a criação de aplicações móveis educativas para *smartphones* Android (TABUENCA *et al.*, 2014).

Dessa forma, sistemas para modelagem de aulas de campo e geração de aplicações ubíquas são importantes ao permitir que os profissionais de ensino se concentrem mais nas questões pedagógicas da aula do que na linguagem de programação. Além disso, diante da variedade de ferramentas e padrões existentes para apoiar os processos de ensino e aprendizagem, a facilidade de sistemas compartilharem entre si formatos comuns de arquivos, torna-se um requisito relevante para promover o seu uso. Por fim, tendo em vista a grande diversidade de dispositivos móveis, a capacidade de geração de aplicações para plataformas distintas constitui-se uma característica importante para esse tipo de software.

Em levantamento bibliográfico realizado, não foi encontrada uma solução que possibilitasse de forma integrada a execução das seguintes funcionalidades por profissionais com pouco ou nenhum conhecimento sobre programação: modelagem de aulas de campo considerando os aspectos pedagógicos (como fluxo de aprendizagem e perfil dos participantes) com uso das tecnologias ubíquas; e, a partir desses modelos, gerar aplicações para diferentes sistemas operacionais móveis. Além disso, a maioria das soluções similares encontradas não utilizava uma especificação aberta que possibilitasse a interoperabilidade dos modelos elaborados com outras ferramentas.

1.2 Hipótese e questões de pesquisa

Esta tese de doutorado visa investigar a seguinte hipótese:

Um ambiente integrado que possibilita o projeto gráfico de aulas de campo suportadas por recursos de computação ubíqua é capaz de produzir modelos abertos e, a partir deles, gerar aplicações fáceis de usar e úteis para os alunos em campo utilizando dispositivos móveis com diferentes sistemas operacionais.

A partir dessa hipótese, quatro questões de pesquisa (QP) foram levantadas. São elas:

QP01: Quais são os principais requisitos que uma aplicação ubíqua deve ter para auxiliar aulas de campo?

QP02: Como especificar as aulas de campo ubíquas de modo que seja possível a representação de atividades de aprendizagem de maneira estruturada, aberta, semântica e legível via computador?

QP03: É possível desenvolver uma ferramenta de modelagem que possibilite a representação gráfica de aulas de campo ubíquas por profissionais de ensino com pouco ou nenhum conhecimento sobre linguagens de programação?

QP04: Como possibilitar a geração automática de aplicações fáceis de usar, úteis e compatíveis com diferentes sistemas operacionais, a partir das especificações das aulas de campo ubíquas?

1.3 Objetivo principal e metas

O principal objetivo desse trabalho é desenvolver e avaliar um ambiente computacional, composto por uma linguagem de modelagem e ferramentas para favorecer a utilização das tecnologias ubíquas em aulas de campo. Esse ambiente é denominado Ubiquitous Field Classes Inventor (UFC-Inventor). Para se alcançar esse objetivo, as seguintes metas precisaram ser alcançadas:

Meta 01: Identificar os requisitos essenciais de aplicações ubíquas voltadas para apoiar aulas de campo.

Meta 02: Elaborar um modelo para especificação das aulas de campo ubíquas que seja compreensível por computador, aberto e permita a representação de diferentes estratégias pedagógicas.

Meta 03: Construir ou evoluir uma ferramenta de modelagem gráfica para representação das atividades de aprendizagem e do fluxo em que elas acontecem em uma aula de campo ubíqua, que permita a conversão para o modelo de especificação citado na Meta 02.

Meta 04: Desenvolver um módulo conversor que tenha como entrada uma especificação de aula de campo ubíqua, seguindo o modelo estabelecido na Meta 02, e permita a geração de uma aplicação ubíqua para diferentes sistemas operacionais, conforme características estabelecidas no modelo.

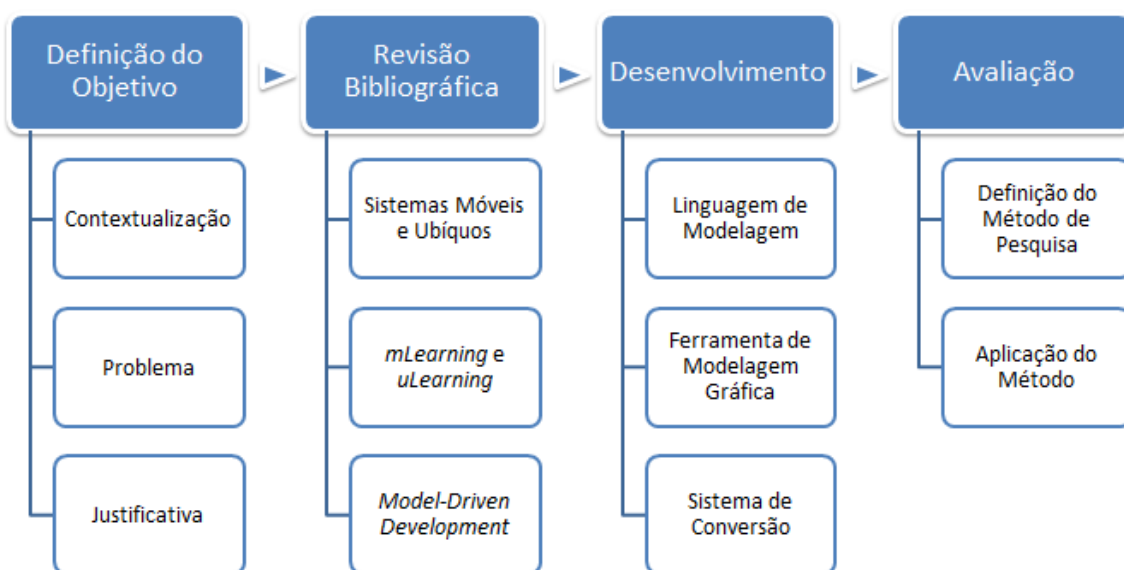
Meta 05: Avaliar o ambiente como um todo, desde o uso da ferramenta de modelagem, passando pelo teste do funcionamento do processo completo de geração das aplicações para diferentes sistemas operacionais móveis, até a análise da usabilidade em campo com alunos em uma aula real.

1.4 Metodologia

Para se desenvolver o UFC-Inventor e alcançar as metas citadas, a metodologia adotada nesse trabalho seguiu as recomendações propostas por Wazlawick (2009), que preveem as seguintes etapas: definição do objetivo da pesquisa, revisão bibliográfica, desenvolvimento do objetivo proposto e aplicação de método científico para avaliar os resultados alcançados.

A Figura 1 apresenta uma visão geral da metodologia utilizada, destacando os principais aspectos. O primeiro passo desse trabalho consiste na definição do objetivo da pesquisa. As seções 1.2 e 1.3 apresentaram a contextualização sobre o tema alvo da pesquisa, os problemas que se pretende solucionar e as justificativas para a importância do ambiente proposto. A seção 1.3 descreve o objetivo principal e as metas desse trabalho.

Figura 1 – Visão Geral da Metodologia



Fonte: elaborada pelo autor.

A segunda etapa consiste no estudo dos temas que compreendem a fundamentação teórica necessária para o desenvolvimento desta tese, que são: desenvolvimento de sistemas ubíquos, desenvolvimento dirigido por modelos, os conceitos de *Mobile learning* e *Ubiquitous learning*, ferramentas de autoria de aplicações educativas móveis e modelagem de processos e recursos educacionais.

Considerando-se a definição dos objetivos e os conhecimentos adquiridos com a revisão bibliográfica, parte-se para o desenvolvimento da contribuição da pesquisa em si. Nesta etapa serão desenvolvidos: a linguagem de modelagem para especificação das aulas de campo

ubíquas; a ferramenta para representação das atividades de aprendizagem e do fluxo em que elas acontecem em uma aula de campo ubíqua através de modelos gráficos e que permita a conversão para a linguagem proposta; e, o sistema conversor que realize a transformação da especificação de uma aula de campo ubíqua na linguagem proposta para uma aplicação ubíqua.

Finalmente, após o desenvolvimento do objetivo principal e metas propostas, os resultados alcançados serão avaliados. Para isso, um método de validação científico deve ser adotado e aplicado. Essa etapa envolve a seleção de usuários que realizarão testes com os softwares desenvolvidos e, posteriormente, responderão os instrumentos de avaliação.

1.5 Organização do documento

Neste capítulo de introdução, foram apresentados os fatores motivadores para a esta tese de doutorado, a hipótese e as questões de pesquisa envolvidas, o objetivo principal e as metas desta pesquisa, como também a metodologia que é utilizada para se alcançar os fins propostos.

O restante deste documento está organizado da seguinte maneira:

Capítulo 2 – O segundo e próximo capítulo desta tese aborda os principais assuntos que representam a base teórica: as características das aulas de campo, os conceitos relacionados com o desenvolvimento de sistemas ubíquos, o paradigma do desenvolvimento dirigido por modelos, os conceitos de *Mobile learning* e *Ubiquitous learning* e a modelagem de processos e recursos educacionais.

Capítulo 3 – No terceiro capítulo são descritos trabalhos acadêmicos mais relevantes que apresentam soluções relacionadas com esta tese de doutorado. Uma análise comparativa entre os trabalhos relacionados também é apresentada.

Capítulo 4 – O quarto capítulo é dedicado a apresentar o mapeamento sistemático realizado para identificar os requisitos essenciais para sistemas ubíquos aplicados a aulas de campo.

Capítulo 5 – O quinto capítulo é dedicado a mostrar o ambiente desenvolvido nesta tese de doutorado: o ambiente UFC-Inventor. Além de uma visão geral do ambiente, os seus componentes são detalhados: a linguagem ML4UL e os módulos UFC-GLM e UFC-Generator.

Capítulo 6 – No sexto capítulo são apresentados os detalhes do processo de avaliação do ambiente UFC-Inventor, destacando-se: a metodologia, os participantes e o procedimento para realização do estudo de caso e da avaliação de usabilidade do UFC-Inventor.

Capítulo 7 – No sétimo capítulo são listados os resultados alcançados ao longo do curso de doutorado, as principais contribuições desta tese, os artigos científicos produzidos e os trabalhos futuros oriundos dessa pesquisa.

Após os capítulos com o conteúdo desta tese, são apresentadas as referências bibliográficas utilizadas e os anexos citados ao longo do texto. O próximo capítulo apresenta a fundamentação teórica desta pesquisa de doutorado.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os principais assuntos que representam a base teórica desta tese de doutorado. A Seção 2.1 apresenta as características e a importância das aulas de campo. Na Seção 2.2 são apresentados conceitos fundamentais relacionados com o desenvolvimento de sistemas e ubíquos. Na Seção 2.3, o paradigma do desenvolvimento dirigido por modelos é detalhado, com a descrição de tópicos essenciais, como as linguagens de modelagem e as transformações entre modelos. A Seção 2.4 apresenta os conceitos de *Mobile Learning* e *Ubiquitous Learning*. Por fim, a Seção 2.5 aborda a modelagem de processos e recursos educacionais, com destaque para a especificação IMS Learning Design.

2.1 Aulas de Campo

A aula de campo é uma prática educativa importante e utilizada em diversas áreas, seja do ensino fundamental, médio ou superior. As aulas de campo possibilitam uma abordagem ao mesmo tempo mais complexa e menos abstrata dos fenômenos estudados em diferentes domínios do conhecimento (ROSLIN *et al.*, 2013). A

Figura 2 mostra diferentes tipos de aulas de campo, onde é possível observar os estudantes visitando locais abertos e espaços fechados para estudar, por exemplo, a vegetação aquática ou uma exposição em um museu. As aulas de campo também promovem o espírito científico dos alunos por meio do aumento da capacidade de observação e descoberta em diversas áreas, que vão desde matérias do ensino fundamental, como ciências, ecologia e história, até disciplinas do ensino superior, como Geologia Geral.

Diversos pesquisadores destacam os benefícios das aulas de campo para a aprendizagem dos alunos, por exemplo: a aula de campo é essencial, pois através dela é possível identificar de fato o que é estudado na sala de aula e perceber as diversas interações do homem com o meio ambiente (FIGUEIREDO; SILVA, 2009); e as aulas de campo fornecem experiências reais e ampliam as possibilidades de aprendizagem através da expansão do mundo conhecido pelos alunos (NABORS; EDWARDS; MURRAY, 2009).

Segundo Shakil, Faizi e Hafeez (2011), uma aula de campo é dividida em três fases: a pré-aula de campo (pré-campo), que corresponde à fase de planejamento e preparação para aula de campo; a segunda fase, que é aula de campo em si; e a pós-aula de campo (pós-

campo), que acontece posteriormente ao término da aula em campo e envolve a análise dos dados coletados.

Figura 2 – Exemplos de aulas de campo



Fonte: imagens capturadas na Internet e editadas pelo próprio autor.

A aula de campo permite aos alunos relacionar vários conceitos, valores, e procedimentos aprendidos em sala de aula às suas práticas cotidianas. Enquanto ela acontece, os estudantes precisam realizar diferentes atividades, tais como observar o ambiente, fazer anotações, realizar medições, além de escutar as orientações e interagir com o professor. Para Behrendt e Franklin (2014), o sucesso de uma aula de campo está em dois fatores relacionados ao professor: boa preparação da aula e interação eficiente com os alunos em campo. Nesse sentido, as tecnologias ubíquas têm se mostrado como importantes ferramentas para auxiliar os alunos durante as atividades de aprendizagem nas aulas de campo (CHIANG; YANG; HWANG, 2014; CHIN; CHEN, 2013; HWANG; TSAI, YANG, 2008).

2.2 Desenvolvimento de sistemas ubíquos

Weiser introduziu o termo computação ubíqua para se referir às tecnologias invisíveis que se misturam com o ambiente, o qual faz a seguinte afirmação: “As tecnologias mais profundas são aquelas que desaparecem. Elas se misturam com os objetos do dia a dia até se

tornarem indistinguíveis no ambiente” (WEISER, 1991). Os sistemas desenvolvidos para os ambientes ubíquos devem ser projetados para auxiliar as pessoas de forma proativa, transparente e casual na execução das suas atividades cotidianas (ROCHA, 2013).

Um termo frequentemente relacionado à computação ubíqua é a computação pervasiva, que segundo (HANSMANN, 2003) pode ser definida como a capacidade de fornecer acesso a serviços de forma conveniente quando e onde o usuário necessitar. De acordo com (NIEUWDORP, 2007), vários autores têm apresentado pesquisas em paralelo com conceitos próximos para computação ubíqua e computação pervasiva, com vários aspectos comuns e intercambiáveis, o que, segundo esse próprio autor, gera confusão entre os termos.

Para fins de padronização, na próxima seção são apresentadas características comuns presentes tanto em sistemas ubíquos quanto em sistemas pervasivos. Além disso, neste documento, será utilizado o termo sistema ubíquo para referenciar o software que atenda a essas características, independentemente se na pesquisa original ele foi referenciado como ubíquo ou pervasivo.

Além disso, neste documento, consideram-se como sistemas ubíquos aqueles que incluem necessariamente a utilização de dispositivos móveis. Dessa forma, não estão incluídos no escopo deste trabalho sistemas que implementam recursos da computação ubíqua sem a utilização de dispositivos móveis pelos usuários, como no artigo sobre casas inteligentes apresentado por Sun *et al.* (2013). Com isso, ao longo do documento, as referências a sistemas e tecnologias ubíquas assumem a inclusão dos aspectos das tecnologias móveis.

Considerando o surgimento de softwares ubíquos com diversos fins, os pesquisadores também têm proposto arquiteturas variadas que atendam às necessidades específicas desses sistemas, por exemplo: para ambientes ubíquos de ensino e aprendizagem (WANG; WU, 2011) e para dar suporte à construção de aplicações no escopo de *Smart Cities* (MACIEL *et al.*, 2014).

O desenvolvimento de arquiteturas e sistemas que atendam às características da computação ubíqua impõe aos engenheiros de software alguns desafios, tais como: heterogeneidade dos dispositivos, ausência de estrutura fixa, dinamicidade do ambiente, suporte à mobilidade dos usuários, descoberta dos serviços e segurança das informações e recursos compartilhados (CASSOU *et al.*, 2009; KINDBERG; FOX, 2002; MAIA; ROCHA; ANDRADE, 2009). Para contorná-los, diferentes abordagens têm sido utilizadas, com

destaque para: engenharia dirigida por modelos, *middlewares*, *frameworks*, linhas de produto de software e padrões de projeto.

Neste capítulo, são apresentados os conceitos fundamentais relacionados ao desenvolvimento de sistemas ubíquos, as características comuns presentes neles, tipos de arquiteturas usadas e as principais abordagens de desenvolvimento que estão sendo utilizadas para implementá-los.

2.2.1 Características dos sistemas ubíquos

Em um ambiente de computação ubíqua, serviços e facilidades computacionais são disponibilizados para os usuários de forma espontânea, transparente e através do uso de interfaces naturais embutidas em dispositivos do cotidiano. Considerando isso, múltiplas implementações de ambientes ubíquos podem ser desenvolvidas nem sempre seguindo a mesma abordagem ou paradigma. Por exemplo, diversos pesquisadores têm construído sistemas ubíquos com arquiteturas distintas (BERTRAN *et al.*, 2014; DLUDLA *et al.*, 2013; JEONG; JOO; JEONG, 2013). As próximas seções apresentam as características dos sistemas ubíquos, os elementos essenciais presentes neles e alguns exemplos de implementações.

Spínola e Travassos (2012) apresentam uma revisão sistemática sobre ambientes de computação ubíqua. Eles partiram de um levantamento inicial com 751 artigos que, após a aplicação de filtros, resultaram em 41 selecionados. Entre outros resultados, essa revisão identificou 10 características principais encontradas em sistemas ubíquos, que são descritas a seguir.

- i. **Onipresença dos serviços.** Permite aos usuários se deslocarem no ambiente ubíquo e poderem utilizar diferentes serviços independentemente do local e momento. Por exemplo, em um museu, o visitante ao se deslocar pelas diferentes salas pode receber as informações sobre as obras de arte de cada um dos espaços que está visitando no momento que está entrando, através de um *smartphone* conectado à rede do museu. Ao se aproximar de uma obra específica (como uma pintura ou escultura) poderá receber informações detalhadas sobre ela.
- ii. **Invisibilidade.** Capacidade de entregar os serviços computacionais aos usuários de forma transparente, de forma a diminuir a sensação de uso explícito das tecnologias. Por exemplo, um ambiente que fica monitorando as condições do ambiente e ajustando o condicionador de ar ou solicitando manutenção, sem a intervenção do usuário.

- iii. **Sensibilidade ao contexto.** Consiste na capacidade do sistema ubíquo para coletar informações contextuais, que estão relacionadas com a localização do usuário, as condições do ambiente que o usuário se encontra, o momento da interação, entre outras. Segundo Dey e Abowd (1999), contexto é toda e qualquer informação que possa ser usada para caracterizar uma entidade (pessoa, local ou objeto) relevante para a interação entre o usuário e o sistema. Por exemplo, em um museu, à medida que o visitante se desloca pelos espaços, o sistema ubíquo registra a localização das salas por onde passou e as obras as quais pediu informação, anotando assim as preferências dos visitantes.
- iv. **Comportamento adaptativo.** Capacidade do sistema ubíquo em ajustar-se para oferecer serviços computacionais de acordo com informações contextuais observadas (e.g., perfil do usuário, dispositivo de acesso, localização, estado corrente do sistema). Esta característica está diretamente relacionada à sensibilidade do contexto, que captura, interpreta, infere e caracteriza o contexto do usuário ao acessar o sistema. Por exemplo: em um museu, após o sistema ubíquo obter a localização do visitante e identificar em que sala ele está entrando, o sistema sugere informações (textos, imagens e vídeos) sobre aquele espaço.
- v. **Captura de experiência.** Característica que torna os sistemas ubíquos capazes de capturar e registrar as experiências relacionadas às interações entre o usuário e o ambiente em que ele se encontra. Por exemplo, o sistema em um museu pode aprender quais as preferências de obras/estilos de exposição que o usuário mais interagiu ou buscou informação nas salas anteriores para poder recomendar em novas salas com instalações similares.
- vi. **Descoberta de serviços.** Mecanismo que suporta a descoberta proativa de serviços disponíveis no sistema ubíquo de acordo com as condições do ambiente. Por exemplo: ao chegar a um museu, o aplicativo no *smartphone* do visitante interage com o sistema do museu recebendo as informações sobre as exposições que estão disponíveis naquele momento.
- vii. **Composição de funções.** Capacidade de criar um serviço demandado pelo usuário através da composição de outros serviços existentes no sistema ubíquo. Por exemplo, um usuário precisa registrar informações geolocalizadas (com as coordenadas de latitude e longitude), porém não conta com o recurso do GPS no seu *Smartphone*. O sistema identifica outros dispositivos que tenham esse serviço e o disponibilizam.

- viii. Interoperabilidade espontânea.** Característica que permite ao usuário utilizar serviços de diferentes dispositivos presentes no ambiente ubíquo de forma transparente. Por exemplo: em um museu, o usuário pode pausar um vídeo no celular e continuar assistindo a partir do instante em que parou através de uma lousa interativa disponível.
- ix. Heterogeneidade de dispositivos.** Capacidade de o sistema ubíquo ser composto por dispositivos com diferentes características, por exemplo: diferentes sistemas operacionais, como Android, iOS e Windows Phone; se é móvel ou fixo; diferentes tipos de conexão, como WiFi ou bluetooth ou via cabo.
- x. Tolerância a falhas.** Característica do sistema ubíquo que permite que ele se adapte em caso de falhas no ambiente. Por exemplo, o recurso de GPS do dispositivo do usuário deixa de funcionar, então o sistema ubíquo localiza um dispositivo próximo com esse recurso funcionando, habilita-o e disponibiliza ao usuário sem a necessidade de configurações adicionais por parte dos usuários.

Conforme constatado pelos autores, é importante destacar que um sistema para ser considerado ubíquo não precisa necessariamente implementar todas as características listadas (SPÍNOLA; TRAVASSOS, 2012). Os autores mostram em seu estudo que nenhum dos artigos levantados demonstrou suporte a mais que 65% dessas características.

2.2.2 Componentes presentes em sistemas ubíquos

Machado et al. (2013) fizeram uma revisão sistemática para identificar as principais partes (elementos, serviços e módulos) que compõem os sistemas ubíquos. Eles fizeram um levantamento de 179 artigos na área e identificaram um conjunto com onze componentes essenciais para softwares de computação ubíqua, que são descritos a seguir.

- i. Sensor.** Elemento de hardware responsável por fornecer a informação de contexto.
- ii. Atuador.** Elemento de hardware responsável por mudar o ambiente ubíquo, dando o retorno para o usuário.
- iii. Serviço de contexto.** Serviço computacional disponibilizado pelo ambiente para recuperar a informação de contexto dos sensores.
- iv. Serviço de atuação.** Serviço computacional que informa ao atuador qual ação de mudança deve ser executada no ambiente.
- v. Repositório de contexto.** Banco de dados que registra as informações contextuais e parâmetros de qualidade do ambiente ubíquo.

- vi. **Módulo de eventos.** Elemento responsável por monitorar a ocorrência de eventos assíncronos no ambiente.
- vii. **Módulo de raciocínio.** Elemento responsável por produzir novas informações de contexto a partir dos dados existentes.
- viii. **Módulo de adaptação.** Elemento responsável por mudar o comportamento do sistema de acordo com regras pré-configuradas.
- ix. **Mecanismo de acoplamento e mobilidade.** Mecanismo responsável por implementar de forma transparente a busca de serviços e a comunicação entre os dispositivos móveis existentes no ambiente.
- x. **Módulo de composição.** Elemento responsável pela composição das informações a partir do repositório de contexto.
- xi. **Módulo de segurança.** Elemento responsável por implementar as regras de proteção e tolerância a falhas do sistema ubíquo.

Da mesma forma que as características citadas na seção anterior, um sistema não precisa implementar todos esses elementos para se considerado ubíquo. Na próxima subseção, são apresentados exemplos que demonstram a implementação de alguns desses elementos em sistemas ubíquos.

2.2.3 *Exemplo de implementação de sistema ubíquo*

Lima *et al.* (2013) apresentam um sistema ubíquo que funciona como um guia de visitas do laboratório de pesquisas GREat¹: o GREat Tour. Esse sistema consiste em uma aplicação sensível ao contexto que utiliza os sensores do *smartphone* para fornecer a localização do visitante no laboratório, através de duas formas: NFC² e QR-code (código de barra Quick Response).

A Figura 3 apresenta os componentes da arquitetura do Great Tour. O Gerenciador de Contexto é responsável por armazenar e coordenar operações relacionadas com o contexto do usuário. Este componente recebe, trata e salva informações capturadas pelos sensores do Dispositivo Móvel (DM). As informações de contexto capturadas pelo Gerenciador de Contexto englobam, além das informações dos sensores, informações do próprio dispositivo, como o nível da carga da bateria e o tipo de rede (3G ou Wi-Fi) que está sendo utilizado.

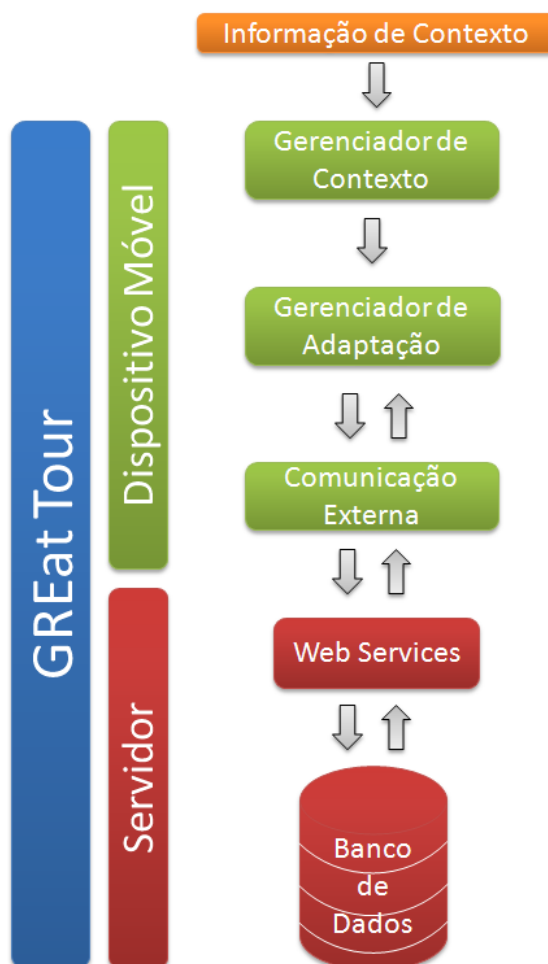
¹ GREat: Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREAT, 2016).

² NFC (*Near Field Communication*) é uma tecnologia que permite a troca de informações entre dispositivos sem a necessidade de cabos ou fios (ATOJI, 2010).

Com base nas informações de contexto coletadas, o Gerenciador de Adaptação efetua as adaptações na aplicação. O banco de dados da aplicação, por outro lado, disponibiliza informações do laboratório GREat para a aplicação, como os mapas do ambiente, os endereços web das imagens e dos vídeos e informações das pessoas que trabalham em cada sala.

A comunicação entre a aplicação móvel e o banco de dados é intermediada por *web services*, utilizados para isolar a parte móvel do GREat Tour da realização de consultas ao banco, facilitando a manutenção e a legibilidade de códigos de conexão com o banco de dados. Dessa forma, a parte do GREat Tour que executa no dispositivo móvel também apresenta um componente responsável por estabelecer e manter essa comunicação.

Figura 3 – Visão do sistema GREat Tour



Fonte: Lima *et al.* (2013).

As arquiteturas de ambientes de computação ubíqua envolvem diferentes elementos, como sensores e atuadores, e serviços específicos, como sensibilidade ao contexto, adaptação

e segurança. Os desenvolvedores precisam se preocupar com diversos aspectos, desde a heterogeneidade dos dispositivos que utilizarão o sistema ubíquo até o controle de acesso aos recursos compartilhados. Além disso, as aplicações precisam ser desenvolvidas de forma que forneçam serviços personalizados e voltados ao contexto do usuário.

2.2.4 Abordagens para o desenvolvimento de sistemas ubíquos

Para facilitar o desenvolvimento de sistemas ubíquos e proporcionar ganhos de produtividade e qualidade, diferentes abordagens têm sido utilizadas. Preuveneers e Novais (2012) mostram uma pesquisa que compara diferentes metodologias para a construção de sistemas ubíquos baseadas no conceito de reuso de software: engenharia dirigida por modelos, *middlewares*, linhas de produto de software e padrões de projeto. Eles concluem afirmando que o desenvolvimento de sistema ubíquos não é uma tarefa trivial e que para implementá-los, faz-se necessária a adoção de uma metodologia de reuso de forma sistemática, que proporcione benefícios tanto em tempo de projeto como de execução também.

Nas próximas seções são apresentadas visões gerais das principais estratégias da engenharia de software aplicadas ao desenvolvimento de sistemas ubíquos.

2.2.4.1 Engenharia dirigida por modelos

A engenharia dirigida por modelos (em inglês, *Model-Driven Engineering* – MDE) utiliza modelos e transformações sobre eles para gerar outros modelos e até código compilável de aplicações (maiores detalhes a respeito de MDE na Seção 2.3). Estudos têm sido desenvolvidos no sentido de aproveitar os benefícios da *Model-Driven Engineering* para construção de sistemas ubíquos. Por exemplo, Tesoriero *et al.* (2010) apresentam uma metodologia para desenvolvimento de sistemas ubíquos que agrupa os modelos em três camadas que seguem a especificação da MDA (*Model Driven Architecture*).

Vários estudos em MDE têm utilizado a UML (*Unified Model Language*) como linguagem para representação dos seus modelos. Por exemplo, Serral, Valderas e Pelechano (2010) apresentam uma linguagem de modelagem baseada em UML para representação de sistemas ubíquos sensíveis ao contexto com alto grau de abstração: a PervML.

2.2.4.2 Middlewares

Outra importante técnica bastante utilizada para o desenvolvimento de sistemas ubíquos são os *middlewares*, que se caracterizam por intermediar a comunicação entre camadas de software possibilitando a troca de informações entre diferentes componentes de sistemas. Inúmeras pesquisas demonstram os benefícios da utilização de *middlewares* na computação ubíqua. Eles podem facilitar o tratamento da heterogeneidade em redes de sensores, simplificar a portabilidade de aplicações móveis para outras plataformas, e auxiliar o desenvolvimento de aplicações distribuídas sensíveis ao contexto.

Um exemplo de *middleware* para sistemas ubíquos, voltado para a aquisição autoadaptativa de informações contextuais, é apresentado por Maia *et al.* (2013): LoCCAM (*Loosely Coupled Context Acquisition Middleware*). Os autores propõem um modelo para publicação e notificação de informações contextuais baseado em espaço de tuplas. Dessa forma, ele possibilita um acoplamento menor entre aplicações e a camada de aquisição de contexto.

O LoCCAM é uma infraestrutura de gerenciamento de contexto que adota o dispositivo móvel como ponto central de aquisição do contexto e de tomada de decisão adaptativa. A arquitetura desse *middleware* está dividida basicamente em três módulos principais: 1) Os componentes de aquisição de contexto construídos sobre a plataforma OSGi (*Open Services Gateways initiative*); 2) Um mecanismo de coordenação baseado em espaço de tuplas (SysSU); 3) O gerenciador de autoadaptação e ciclo de vida. Os CACs (*Context Acquisition Components*) são os elementos responsáveis por capturar o contexto.

Como apresentado nesta seção, os *middlewares* podem ser utilizados para apoiar a implementação de sistemas ubíquos, entre outras razões, ao facilitar a aquisição e o processamento das informações contextuais. Entretanto, Cassou *et al.* (2012) afirmam que uma desvantagem dos *middlewares* é a implementação com escopo limitado a determinadas funcionalidades. Dessa forma, certos requisitos de sistemas ubíquos não são atendidos, por exemplo: Donohoo, Ohlsen e Pasricha (2011) apresentam um *middleware* que realiza o tratamento de contexto, porém não implementa o gerenciamento de serviços, demandando do engenheiro de software a complementação do código-fonte, além da integração com o *middleware*.

2.2.4.3 Linhas de produto de software

As linhas de produto de software (LPS) constituem uma técnica da engenharia de software para o desenvolvimento de aplicações baseado em artefatos reusáveis (VAN DER LINDEN; SCHMID; ROMMES, 2007). As LPS são voltadas para otimizar a construção de aplicações que apresentem um conjunto comum de funcionalidades que atendam as necessidades específicas de um segmento específico. As principais características comuns e variáveis de uma linha de software são representadas pelo modelo de características (em inglês, *feature model*).

Marinho *et al.* (2013) propõem uma linha de produto de software para produção de guias de visitas móveis e sensíveis ao contexto que são executados nos dispositivos móveis dos visitantes: a MobiLine. Durante as visitas, esses guias podem ter suas funcionalidades adaptadas de acordo com o contexto do ambiente e do próprio visitante, incluindo a localização dele, perfil, características do dispositivo móvel utilizado e informações sobre outros usuários presentes no mesmo espaço.

De acordo com Preuveneers e Novais (2012), o principal benefício das linhas de produto de software para o desenvolvimento de ambientes de computação ubíqua é a documentação explícita da variabilidade de uma determinada família de sistemas. Para eles, as LPS obrigam os desenvolvedores a pensar sobre o quê, por que e como os elementos do software variam. Assim, as aplicações podem de forma mais confiável adaptar-se a plataformas heterogêneas e estarem preparadas para mudanças e circunstâncias não previstas.

Porém, como sinalizado em Thüm *et al.* (2014), as arquiteturas das linhas de produto de software tornam-se complexas e isso demanda a necessidade de um programador especialista na linha, o que pode representar custos a mais ao projeto. Além disso, Thüm *et al.* (2014) afirmam que as LPS requerem métodos específicos para processos da engenharia de software, como a análise e os testes, não sendo possível a utilização de técnicas tradicionais. Isso demonstra a necessidade de não apenas um profissional, mas uma equipe especialista na construção de sistemas baseada em Linhas de Produto de Software.

2.2.4.4 Outras abordagens

Os *frameworks* de programação têm sido aplicados no domínio da computação ubíqua para facilitar o desenvolvimento do software ao elevar o nível de abstração. Por exemplo, o OPEN é um *framework* para prototipagem rápida, compartilhamento e personalização de aplicações sensíveis ao contexto (GUO; ZHANG; IMAI, 2011).

Já a Programação Orientada a Aspectos é uma técnica da engenharia de software que pode ser aplicada ao desenvolvimento de sistemas ubíquos para modularização de partes importantes do código-fonte. Por exemplo, Munnelly, Fritsch e Clarke (2007) apresentam uma abordagem orientada a aspectos para modularização da implementação do contexto do usuário, obtendo melhorias na compreensão, capacidade de gerenciamento e manutenção do código-fonte.

Padrões de projetos também constituem uma técnica da engenharia de software que pode ser aplicada no desenvolvimento de sistemas ubíquos para fornecer soluções repetíveis a problemas que ocorrem recorrentemente na construção de sistemas. Chung *et al.* (2004) propõem uma linguagem para sistemas ubíquos composta por 45 padrões, agrupados em quatro grupos: Gêneros de computação ubíqua, que descrevem as categorias das aplicações ubíquas; Espaços físicos-virtuais, que analisam a forma como os objetos e espaços físicos podem ser mescladas com o virtual dentro do ambiente ubíquo; Privacidade, que descrevem as políticas e os mecanismos de gestão de privacidade do usuário final; e Interação, que detalham as técnicas de interação entre os sensores e os dispositivos.

2.2.5 Desafios para o desenvolvimento de sistemas ubíquos

Os sistemas ubíquos podem ser aplicados em diferentes áreas e construídos através de metodologias distintas, conforme discutido anteriormente. Dessa forma, o engenheiro de software deve estar atento a problemas que podem surgir durante seu desenvolvimento, como os apontados por Bellavista *et al.*(2012), Cassou *et al.*(2012), Maia, Rocha e Andrade (2009) e Santos *et al.*(2013).

Uma das dificuldades está relacionada com a heterogeneidade dos dispositivos, tanto em termos do hardware do equipamento quanto com relação ao sistema operacional. Dependendo da situação, pode ser que seja necessário que o desenvolvedor precise desenvolver um software ubíquo multiplataforma, o que implicará em diferentes implementações das mesmas funcionalidades para cada uma das plataformas-alvo.

Outra questão é relativa à ausência de estrutura fixa dos ambientes, que tem dois efeitos principais: um com relação à usabilidade do aplicativo, que deve considerar que os usuários podem estar em movimento e precisam de interfaces intuitivas; e outro com relação à conectividade com a Internet, que pode não existir e o sistema deve persistir as informações localmente.

Outro obstáculo durante a construção de softwares ubíquos está relacionado à segurança, tanto das informações trafegadas pelo sistema quanto dos recursos compartilhados. Considerando a mobilidade dos usuários e o fato das informações estarem sendo trafegadas em redes sem fio, pode ser necessário que o sistema ubíquo implemente uma camada extra de software com protocolos de segurança.

Tendo em vista o objetivo principal desta tese de doutorado, optou-se por utilizar a metodologia da engenharia dirigida por modelos para possibilitar a geração de aplicações ubíquas a partir de modelos com alto nível de abstração.

2.3 Desenvolvimento dirigido por modelos

As tecnologias da engenharia dirigida por modelos possibilitam a representação dos conceitos do domínio de forma efetiva e diminuem a complexidade das plataformas de programação (SCHMIDT, 2006). Assim, os engenheiros de software podem concentrar-se mais no problema (modelos) que nas linguagens de programação específicas. Além disso, a partir das ferramentas da MDE, podem ser realizadas simulações, tomadas de decisão e geração do código-fonte de aplicações (WILLIAMS *et al.*, 2012).

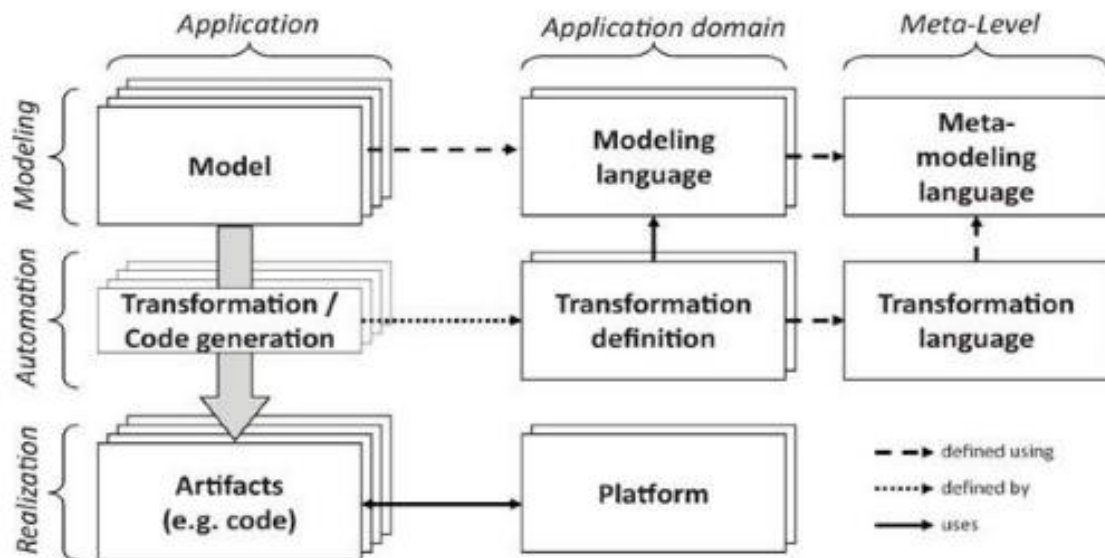
Várias iniciativas têm sido desenvolvidas em MDE. O consórcio internacional OMG (*Object Management Group*) propôs uma abordagem padrão para desenvolvimento de sistemas dirigido por modelos: a MDA – *Model Driven Architecture* (OMG, 2003). O funcionamento da MDA pode ser visto como a aplicação de regras de transformação em um modelo-fonte, especificado, por exemplo, no meta-modelo UML, para obtenção de um modelo-alvo na plataforma destino, por exemplo, Java.

De uma forma geral, a MDE pode ser vista como a aplicação de transformações sobre modelos para a produção de artefatos, que podem ser código-fonte ou outros modelos. A Figura 4 apresenta uma visão geral da engenharia dirigida por modelos. Pode-se observar que para a representação dos modelos existem as linguagens de modelagem e para a realização das transformações, existem as linguagens de transformação.

Do ponto de vista do funcionamento, a MDE pode ser dividida em três níveis: Modelagem, onde os modelos são elaborados; Automação, onde as transformações são realizadas; e o nível Realização, onde são gerados os artefatos finais para execução do sistema. Considerando os modelos utilizados nos processos, a MDE pode ser dividida em três níveis: o nível da Aplicação, que contém os níveis que representam aquela aplicação; Domínio da Aplicação, onde são definidas as linguagens de modelagem e transformações para um domínio específico; e o Metanível, onde estão as definições da metalinguagem e da

linguagem de transformação que são usadas como base para os artefatos do nível de Domínio da Aplicação.

Figura 4 – Visão geral da Engenharia Dirigida por Modelos



Fonte: Brambilla, Cabot e Wimmer (2012).

Um conceito importante na área de engenharia de software dirigida por modelos é o desenvolvimento dirigido por modelos (em inglês *Model-Driven Development - MDD*). Segundo Brambilla, Cabot e Wimmer (2012), o MDD pode ser visto como um subconjunto da MDE, onde modelos são utilizados como artefatos primários para a geração (semi) automática de código-fonte. A MDE é um conceito mais abrangente ao envolver outros processos baseados em modelos além da utilização deles para geração de código-fonte como no MDD, por exemplo: engenharia reversa e gerenciamento da evolução dos modelos de um sistema.

Sistemas de diferentes áreas do conhecimento têm sido construídos utilizando-se as tecnologias de desenvolvimento dirigido por modelos. Por exemplo, Hermida *et al.* (2013) apresentam uma metodologia baseada em MDE para o desenvolvimento de sistemas gerenciais de inteligência empresarial (em inglês *Business Intelligence*). Outro estudo que demonstra os benefícios do uso do desenvolvimento de sistemas dirigidos por modelos é apresentado por Cuadrado, Cánovas Izquierdo e Molina (2014). Eles apresentam casos de sucesso da implantação das técnicas de MDE em duas empresas de software para melhorar os processos de desenvolvimento e manutenção de sistemas.

A engenharia de software dirigida por modelos engloba inúmeros métodos e técnicas, que vão desde a modelagem dos requisitos do sistema até a geração do código-fonte. Nas

próximas seções, são apresentados conceitos importantes relacionados a MDE: linguagens de modelagem, metamodelos, transformações entre modelos, geração do código-fonte e técnicas para o gerenciamento dos modelos.

2.3.1 Linguagens de modelagem

Um dos principais elementos da MDE é a linguagem de modelagem. Através dela, os projetistas especificam os modelos do sistema. Ela possibilita a representação concreta de um modelo conceitual, através de diagramas ou especificações textuais. Existem as linguagens de modelagem de propósito geral, como a UML, e aquelas que são projetadas especificamente para resolver problemas de um determinado domínio: as linguagens específicas de domínio (em inglês Domain-Specific Language - DSL).

Segundo Van Deursen, Klint e Visser (2000), as DSL são especificações executáveis que oferecem funcionalidades importantes focadas em atender um determinado domínio, como a utilização de notações e abstrações apropriadas que modelam de forma clara e concisa os sistemas. Os autores apresentam vantagens e desvantagens para o uso de DLS, que estão resumidas na Tabela 1.

Tabela 1 – Vantagens e desvantagens do uso de DSL.

Vantagens	Desvantagens
As DSL aumentam a produtividade, confiabilidade e portabilidade das soluções.	Os custos para treinamento para utilização das DSL.
Os programas em DSL são concisos e podem ser reusados no mesmo domínio.	A dificuldade para encontrar as DSL apropriadas aos domínios.
As soluções podem ser especificadas no nível de abstração do domínio.	Os custos para desenvolver, implementar e manter uma nova DSL.
As DSL facilitam a conservação e reuso das regras de negócio do domínio.	
As DSL permitem a validação do sistema na etapa de projeto.	

Fonte: Van Deursen, Klint e Visser (2000).

As linguagens específicas de domínio têm sido utilizadas para resolver problemas em diferentes domínios de aplicação. Alguns exemplos de DSL conhecidas são: HTML, para construção de páginas Web; SQL, para especificação de acesso a banco de dados; Logo, para ensino de programação para crianças; MatLab, voltada para resolução de problemas matemáticos, entre outros.

Segundo Brambilla, Cabot e Wimmer (2012), as linguagens de modelagem podem ser definidas a partir de três componentes principais:

- **Sintaxe abstrata.** Descreve a estrutura da linguagem e a forma que as diferentes primitivas podem ser combinadas entre si. A sintaxe abstrata é responsável por informar quais são os elementos da linguagem (como classes, instâncias, atributos) e como eles se relacionam entre si.
- **Sintaxe concreta.** Descreve as representações (textuais ou gráficas) específicas da linguagem de modelagem, englobando as codificações e os aspectos visuais.
- **Semântica.** Descreve o significado dos elementos definidos na linguagem e o significado das diferentes formas de combinação entre eles.

A Tabela 2 utiliza trechos da linguagem de acesso a banco de dados SQL para exemplificar os conceitos de Sintaxe Abstrata, Sintaxe Concreta e Semântica.

Tabela 2 – Exemplos de sintaxes abstrata, concreta e semântica da linguagem SQL.

<i>Sintaxe abstrata</i>	<i>Sintaxe Concreta</i>	<i>Semântica</i>
Instruções: SELECT, DELETE, CREATE, INSERT, etc. Operadores: *, =, >, <, AND, OR, NOT, etc. Expressões condicionais: CASE-WHEN-THEN-ELSE-END	SELECT [DISTINCT] select_list FROM from_clause [WHERE search_condition] [GROUP BY column {, column}[HAVING search_condition]] [ORDER BY column {, column}]	“A instrução SELECT é a base da consulta de qualquer banco de dados SQL. Uma instrução SELECT retorna uma tabela ao cliente que corresponde à consulta.”

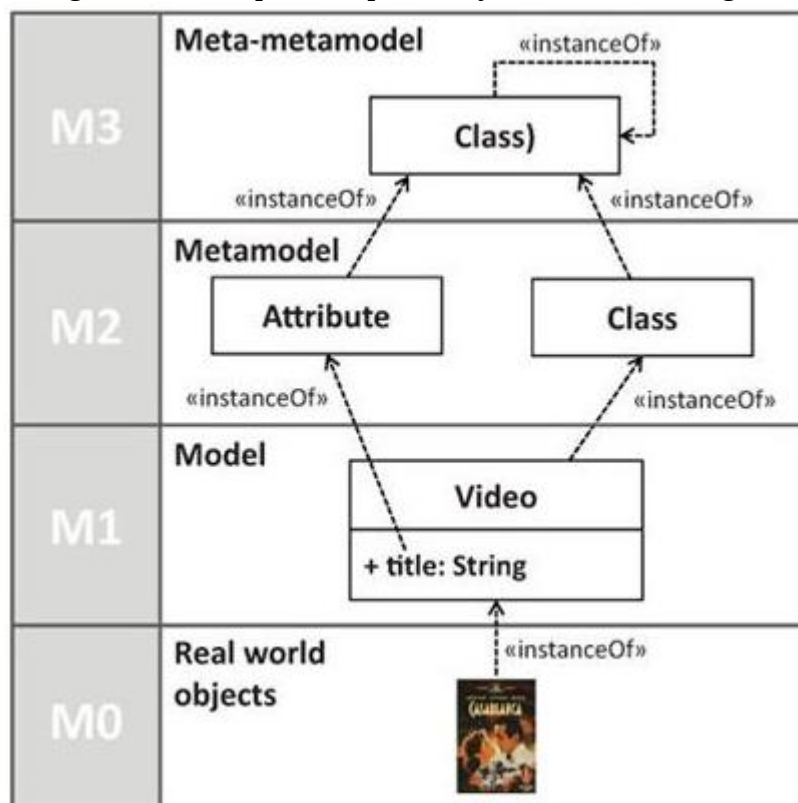
Fonte: elaborada pelo autor.

2.3.2 Metamodelos

Da mesma forma que um modelo pode ser visto como uma abstração dos fenômenos no mundo real, os metamodelos são uma forma de abstração que definem as propriedades desses modelos (BRAMBILLA; CABOT; WIMMER, 2012). Assim, os metamodelos constituem a definição das linguagens de modelagem. Dessa forma, os modelos que definem os modelos são chamados de metamodelos e os modelos que definem os metamodelos são chamados de meta-metamodelos. A Figura 5 mostra um exemplo do funcionamento da metamodelagem, que consiste na especificação dos relacionamentos entre os modelos, metamodelos e meta-metamodelos da linguagem de modelagem.

No exemplo mostrado na Figura 5, os objetos do mundo real (filmes) são representados no nível M0. No nível M1, está o modelo que especifica esses objetos com seus atributos (nesse exemplo, o atributo 'title'). Este modelo tem seus conceitos definidos pelo metamodelo que está no nível M2: Attribute e Class. Finalmente, no nível M3, está o meta-metamodelo que define os conceitos dos metamodelos do nível M2.

Figura 5 – Exemplo de representação da metamodelagem

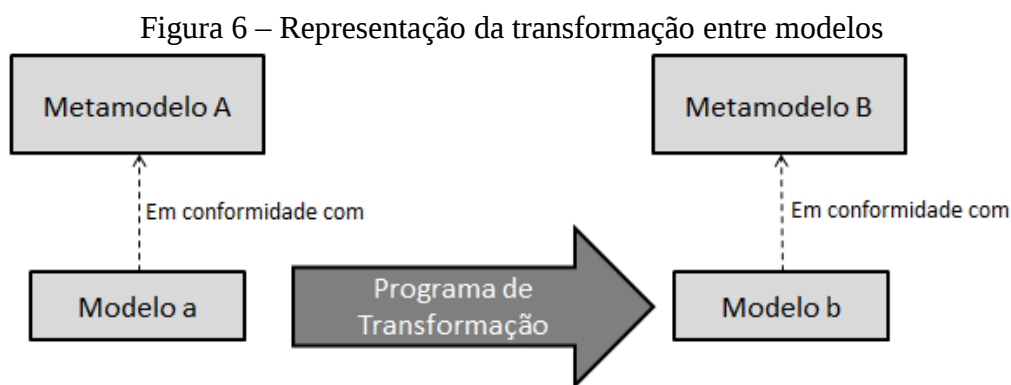


Fonte: Brambilla, Cabot e Wimmer (2012).

2.3.3 Transformações entre modelos

Da mesma forma que os modelos, as transformações entre eles também têm papel fundamental na MDE. As transformações são definidas no metamodelo e especificam os mapeamentos entre diferentes modelos (VÖLTER *et al.*, 2013). As transformações podem ser de modelos entre si (*Model-to-Model*, M2M) ou de modelo para texto (*Model-to-Text*, M2T).

A transformação M2M (*Model-to-Model*) ocorre quando um programa recebe como entrada um ou mais modelos e produz um ou mais modelos como saída. A Figura 6 ilustra como ocorre uma transformação entre modelos. Um exemplo seria transformar um modelo especificado em Diagrama de Classes da UML para um modelo de banco de dados na especificação Diagrama de Entidade e Relacionamento.



Fonte: elaborada pelo autor.

Para a realização das transformações entre os modelos podem ser utilizadas linguagens de transformação. Um exemplo é a ATL (*Atlas Transformation Language*) (JOUAULT *et al.*, 2008). A ATL é uma linguagem de transformação entre modelos baseada no metamodelo MOF (BRAMBILLA; CABOT; WIMMER, 2012). A ATL é composta pelos seguintes elementos: módulos (ATL modules), que descrevem as regras de transformação; *queries*, que definem um modelo para transformação do valor de tipos de dados primitivos; bibliotecas (ATL libraries), que permitem a definição de métodos que poderão ser chamados em diferentes pontos do código-fonte em ATL; tipos de dados, baseados nos tipos das Object Constraint Language – OCL (BRAMBILLA; CABOT; WIMMER, 2012) e, expressões declarativas OCL.

As transformações Model-to-Text (M2T) consistem na derivação de textos a partir de modelos (OLSEN; OLDEVIK, 2007). A principal utilização das transformações M2T é para geração de código-fonte, porém também podem ser usadas para outros fins, por exemplo: produção de documentação, geração de lista de tarefas e construção de casos de testes.

O processo inverso à transformação M2T, isto é, a geração de modelos a partir de textos (Text-to-Model, T2M), também tem papel de destaque na MDE. Sua principal aplicação é a obtenção de modelos gráficos a partir de código-fonte, processo conhecido como engenharia reversa (FAVRE, 2004).

2.3.4 Geração de código-fonte

A geração de código-fonte consiste em um processo completo de transformação Model-to-Text (M2T) produzindo scripts de linguagens de programação que poderão ser compilados para geração de aplicações executáveis. Diferentes estudos apontam as vantagens da geração de código-fonte (BRAMBILLA; CABOT; WIMMER, 2012; KELLY; TOLVANEN, 2008), dentre as quais destacamos:

- Produtividade na construção das aplicações ao reduzir ou eliminar o tempo gasto com a etapa de programação, considerando que os modelos usados já foram aprovados. Além disso, os tempos das etapas de testes e de implantação das aplicações geradas a partir deles também são reduzidos.
- A geração de código-fonte protege a propriedade intelectual do projetista de software, que construiu os modelos do sistema, ao possibilitar a geração de aplicações executáveis sem a necessidade de compartilhar o projeto.
- O código-fonte gerado não segue padrões específicos e pode ser compreendido e mantido por quaisquer desenvolvedores que tenham domínio na linguagem utilizada.
- A geração de código-fonte permite ao projetista de software escolher em que plataforma as aplicações funcionarão. Isso é muito importante porque evita que o cliente fique dependente de um fornecedor de software ou de hardware.
- Necessidades específicas do cliente (como regras de negócio ou questões legais) podem ser modeladas na etapa de projeto e serão incorporadas no código-fonte gerado.
- Trechos de modelos já elaborados podem ser reusados para geração de código-fonte em novos sistemas.
- A geração pode ser flexível, que dizer que apenas partes dos modelos podem produzir código-fonte. Dessa forma, é possível aliar ganhos de produtividade com desenvolvimento específico em trechos do sistema que demandam atenção especial.

Os modelos, metamodelos, linguagens específicas de domínio, transformações Model-to-Model e Model-to-Text são importantes conceitos na área da Engenharia Dirigida por Modelos. Porém, além deles existem outros aspectos importantes envolvidos no gerenciamento do processo de modelagem, tais como: interoperabilidade entre ferramentas de modelagem, testes e verificação de modelos.

2.3.5 Dificuldades ao uso do desenvolvimento dirigido por modelos

O MDD se propõe a aumentar a qualidade dos sistemas, reduzir a complexidade na criação dos programas e promover o reuso de modelos e códigos-fonte. Entretanto, durante o desenvolvimento de projetos de software dirigidos por modelos, alguns obstáculos podem surgir.

Um dos principais problemas está relacionado à manutenção da persistência entre os modelos e seus respectivos códigos-fontes (BRAMBILLA; CABOT; WIMMER, 2012; HAILPERN; TARR, 2006). Um exemplo de situação onde ocorre perda de consistência entre modelo e código-fonte acontece quando são realizadas alterações no corpo do código-fonte e estas não são atualizadas no modelo, tornando-o defasado.

Outro desafio encontrado para adoção do desenvolvimento de software dirigido por modelos está na escolha das ferramentas adequadas ao projeto (DEN HAAN, 2009; MOHAGHEGHI *et al.*, 2009; WHITTLE *et al.*, 2013). Segundo Whittle *et al.* (2013), as ferramentas disponíveis para MDE não são intuitivas, são complicadas de usar e não consideram as condições do ambiente real, estando mais focadas nas próprias funcionalidades do que nos processos. Inclusive, os autores afirmam que o engenheiro de software que pretende trabalhar com uma ferramenta MDD intuitiva deve desenvolver a sua própria.

2.4 Mobile Learning e Ubiquitous Learning

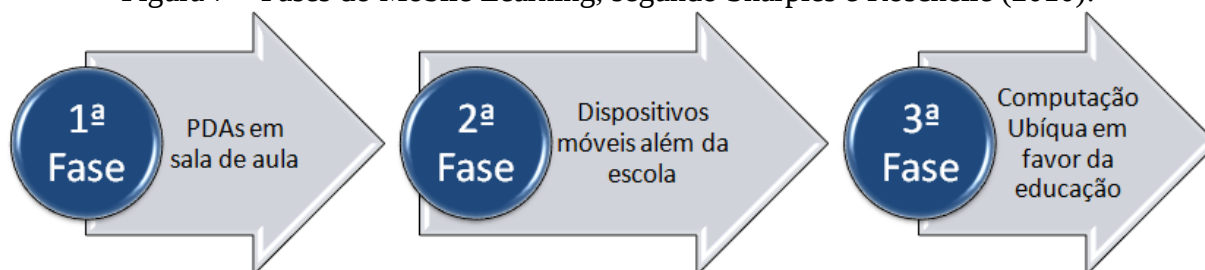
O paradigma *Mobile learning* (ou *m-learning*) surgiu a partir da utilização das tecnologias móveis e sem fio como parte de um modelo de aprendizado integrado (MARÇAL; ANDRADE; RIOS, 2005). Segundo Peng *et al.* (2009), o *m-learning* trata-se de uma evolução do *e-learning* através da adição das tecnologias móveis.

Segundo a classificação cronológica apresentada por Sharples e Roschelle (2010), o *m-learning* pode ser dividido em três fases, conforme pode ser observado na Figura 7. A primeira iniciou-se a partir das iniciativas para inserir os dispositivos móveis (como PDAs –

Personal Digital Assistants) em sala de aula com o objetivo de incrementar os recursos dos professores.

A segunda fase foi caracterizada pela mobilidade, na qual os alunos utilizavam os dispositivos fora de sala de aula. Dessa forma, o alcance do *m-learning* foi expandido através da utilização de aplicações móveis em locais de aprendizagem informal, como museus (MARÇAL; ANDRADE; RIOS, 2005) e zoológicos (YEE *et al.*, 2009). As pesquisas indicavam que os recursos da computação móvel influenciavam positivamente na educação porque estimulavam à aprendizagem mesmo em curtos intervalos de tempo, sem a obrigatoriedade da dedicação de horas como no ensino tradicional (HUMMEL; HLAVACS; WEISSENBOCK, 2002).

Figura 7 – Fases do Mobile Learning, segundo Sharples e Roschelle (2010).



Fonte: Elaborada pelo próprio autor.

Atualmente, estamos na terceira fase do *m-learning*, que é marcada pela utilização dos recursos da computação ubíqua em favor do ensino e aprendizagem. As tecnologias ubíquas permitem, por exemplo, a identificação do contexto dos alunos e, assim, possibilitam a sugestão de conteúdos e atividades de acordo com suas necessidades e situações de aprendizagem. Por exemplo, em um museu, à medida que o aluno se desloca pelos espaços, o sistema identifica a localização dele e oferece informações de acordo com o local onde ele se encontra. Esta é uma das características de sistemas da computação ubíqua conhecida como sensibilidade ao contexto (em inglês *context-awareness*) (SPÍNOLA; TRAVASSOS, 2012). Para essa fase do *m-learning* emprega-se o termo “*ubiquitous learning*” ou “*u-learning*”.

Diversas iniciativas têm sido realizadas no sentido de desenvolver sistemas ubíquos voltados para favorecer os processos de ensino e aprendizagem (IVANOV, 2013; LEE *et al.*, 2012; WU; HWANG; TSAI, 2013). Uma das primeiras e mais referenciadas definições para o termo *u-learning* é apresentada por Ogata e Yoneo (2004), que afirmam que os sistemas de *ubiquitous learning* devem apresentar os seguintes requisitos:

- Permanência. Os alunos não podem perder seus trabalhos e todos os registros deles durante os processos de aprendizagem devem ser mantidos.

- Acessibilidade. Os alunos devem poder acessar o material didático (textos, vídeos e imagens) independentemente da sua localização. Tal informação é acessada de acordo com suas requisições, tornando esse processo de aprendizagem direcionado às necessidades do próprio aluno.
- Immediatez. Os alunos podem recuperar a informação de forma imediata no instante em que eles requisitarem. Dessa forma, questões podem ser solucionadas no momento em que são apresentadas.
- Interatividade. Os alunos podem interagir entre si ou com os professores ou tutores de forma síncrona ou assíncrona.
- Atividades instrucionais localizadas. Os problemas propostos bem como o conhecimento requerido devem ser apresentados de forma natural e considerando a situação dos alunos.

Além de Ogata e Yoneo (2004), outros pesquisadores apresentam definições para *u-learning* (BOYINBODE; AKINTOLA, 2009; COPE; KALANTZIS, 2008). Hwang, Tsai e Yang (2008) demonstram preocupação com a visão de alguns pesquisadores que definem *u-learning* como sendo aprendizagem em qualquer lugar e a qualquer momento, por essa ser uma definição bastante abrangente. Assim, para trabalhar com uma definição menos abrangente, eles utilizam o termo *context-aware u-learning*, para se referir ao uso dos dispositivos móveis com redes sem fio e tecnologias de sensores em atividades de aprendizagem. Nesse trabalho de doutorado será usada essa definição para referenciar o termo *ubiquitous learning*.

Em outro trabalho importante na área, Hwang e Tsai (2011) fizeram um levantamento sobre as pesquisas desenvolvidas em *m-learning* e *u-learning* durante os anos de 2001 e 2010 através da publicação em seis importantes jornais sobre tecnologias na educação. O estudo concluiu que as pesquisas em *m-learning* e *u-learning* tiveram uma forte crescente entre os anos de 2008 e 2010. Em outra conclusão, o estudo aponta que a maior parte dos estudos de caso foi aplicada com alunos do ensino superior (38,3%). Por fim, esse levantamento mostrou que o principal domínio de aplicação foi no ensino de ciências (19,48%), seguido pelo ensino de idiomas (15,58%).

2.4.1 Aulas de campo ubíquas

Desde a década de 1990, os pesquisadores vêm desenvolvendo estudos sobre como utilizar as tecnologias móveis e ubíquas para ampliar os benefícios proporcionados pelas aulas de campo e solucionar os problemas existentes. Diferentes áreas têm experimentado a utilização da computação ubíqua em aulas de campo, por exemplo: em botânica (LO; QUINTANA, 2013), em ecologia (HUNG *et al.*, 2013), e em geografia e arquitetura (ALLWIHAN ; BRAILSFORD; COBB, 2013).

Em 1991, a Apple em parceria com a Orange Grove Middle School utilizou PDAs (Personal Digital Assistants) para aulas de campo sobre o Sabino Canyon, na cidade de Tucson no Arizona, Estados Unidos (GRANT, 1993). Além dos PDAs, os alunos utilizaram câmeras digitais para capturar imagens e vídeos, walkie-talkies para comunicação entre os grupos que estavam espalhados e modem de celular para enviar os dados para a escola que estava a 15 milhas de distância do local estudado. Os autores afirmaram que aconteceram alguns problemas na relação entre o uso das tecnologias móveis versus a prática convencional de estudo utilizada nas aulas de campo, como o consumo da bateria que acabava muito rápido. Eles concluíram que mais estudos seriam necessários para aperfeiçoar o uso das novas tecnologias em conjunto com as modernas teorias de aprendizagem.

No projeto RAFT (*Remote Accessible Field Trips*), foi desenvolvido um sistema (*Mobile Collector*) que permitia aos alunos, nas aulas de campo, anotarem informações, capturarem fotos e buscar informações que poderiam ser reusadas por eles (KRAVCIK *et al.*, 2004). O sistema desenvolvido tinha uma interface que permitia acompanhar tudo que estava sendo registrado pelos alunos em tempo real através da Internet.

Davies *et al.* (2010) desenvolveram uma pesquisa que separou estudantes de geologia em dois grupos: um que ficou no laboratório e outro que foi a campo com telefones celulares. Os participantes do segundo grupo faziam as anotações, tiravam fotos, gravavam vídeos e transmitiam em tempo real (via rede Wi-Fi) para os alunos do primeiro grupo. Os alunos que estavam em campo também podiam solicitar informações aos que estavam no laboratório. Os estudantes utilizavam diferentes aplicações no *smartphone* e necessitavam de conexão com a Internet para trocar informações com o grupo que ficou no laboratório.

Uma pesquisa que une conceitos *u-learning* e inteligência artificial é mostrada em (WU; HWANG; TSAI, 2013). Eles apresentam um sistema especialista que guia os alunos durante as aulas de campo, sugerindo pontos de observação, de coleta de dados e de realização de tarefas de acordo com o perfil e contexto do aluno. Os alunos utilizam leitores e

etiquetas de RFID para identificação das rochas. O sistema ainda conta com um questionário para avaliar os conhecimentos adquiridos pelos alunos durante as atividades de campo.

Nesse trabalho, utilizaremos o termo Aulas de Campo Ubíquas (UFC – *Ubiquitous Field Classes*) para referenciar as práticas educativas realizadas em campo com suporte das tecnologias móveis e ubíquas. Diversas pesquisas têm demonstrado os benefícios proporcionados pelo uso das tecnologias ubíquas em aulas em campo (MARÇAL *et al.*, 2013; LO; QUINTANA, 2013; WU; HWANG; TSAI, 2013), dentre os quais destacamos: identificação do contexto do aluno, através de sensores, para entrega de conteúdos e exercícios de acordo com a situação dele naquele momento; registro das preferências dos estudantes para recomendações futuras; e comunicação entre alunos e professores através de redes sem fio para esclarecimento de dúvidas ou compartilhamento de informações.

2.4.2 Ferramentas de autoria de aplicações educativas móveis

Segundo Flôres, Tarouco e Reategui (2014), as ferramentas de autoria ou sistemas de autoria são softwares voltados para favorecer os processos de ensino e aprendizagem através da facilitação da criação de material educacional. Através delas, os autores (como professores e especialistas de área) podem, por exemplo, criar conteúdos digitais para auxiliar a fixação de conceitos vistos em sala de aula. Os autores podem construir estes conteúdos mesmo sem possuir conhecimentos de programação.

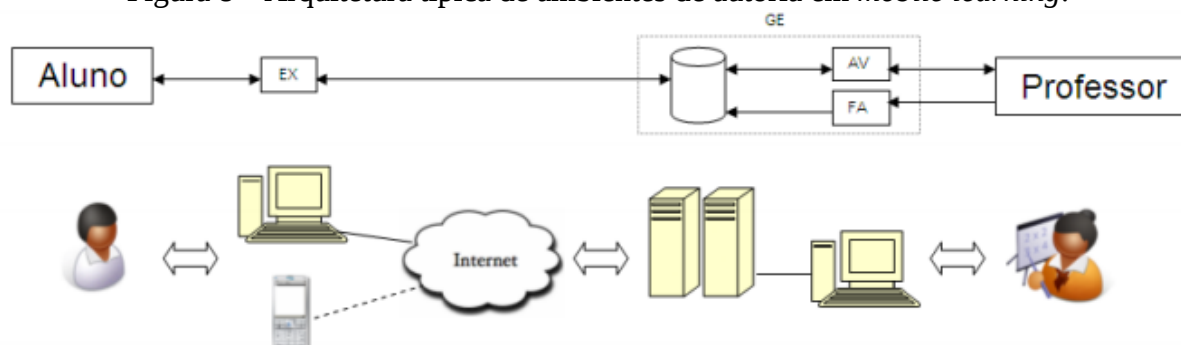
Existem outros tipos de softwares, igualmente conhecidos como ferramentas de autoria, que também possibilitam a geração de aplicações com diversas finalidades, tais como jogos e portais de Internet. Além disso, alguns autores consideram os ambientes de programação como ferramentas de autoria (HAND, 2012). Entretanto, esse documento foca nas ferramentas com propósitos educacionais que possam ser utilizadas por pessoas com pouco ou nenhum conhecimento técnico de programação.

No caso das aplicações de *m-learning*, em geral, as ferramentas de autoria são softwares para plataforma *desktop* que possibilitam que os professores construam e gerem aplicações para dispositivos móveis, conforme mostrado na Figura 8. Nessa figura, pode-se observar a ferramenta de autoria (FA) que faz parte de um módulo gerenciador (GE) juntamente com o módulo de avaliação (AV) que são voltados para o professor, e, do outro lado, o módulo executado pelo aluno (EX).

Segundo Murray, Woolf e Marshall (2004), os principais objetivos das ferramentas de autoria são:

- Diminuição dos esforços necessários para construção de material educacional;
- Redução dos requisitos mínimos necessários que o autor precisa ter para elaborar conteúdo educacional digital;
- Geração dos protótipos de conteúdo ou de aplicativos com maior facilidade.

Figura 8 – Arquitetura típica de ambientes de autoria em *mobile learning*.



Fonte: Souza (2010).

Quando se considera o desenvolvimento para dispositivos móveis, a importância das ferramentas de autoria é ampliada devido à complexidade imposta para construção de aplicativos para essas plataformas. Além dos requisitos postos pelas limitações dos dispositivos móveis (como tamanho da tela, memória interna de armazenamento e velocidade de conexão), as aplicações para esses dispositivos apresentam as seguintes características particulares que aumentam a complexidade do seu desenvolvimento (WASSERMAN, 2010): interação com outras aplicações; utilização de interface multimodal; grande variedade de software e hardware; e elevado consumo da bateria.

Nesse cenário, as ferramentas de autoria voltadas para modelar e gerar aplicações móveis com fins de aprendizagem têm grande importância na tentativa de se transformar os dispositivos móveis em recursos de ensino para o professor (GIEMZA; BOLLEN; HOPPE, 2011; LIMA *et al.*, 2011; ISHITANI, 2012; CHIN; CHEN, 2013).

2.5 Modelagem de processos e recursos educacionais

Existem diferentes iniciativas voltadas à padronização de recursos e processos de ensino e aprendizagem através do computador, que envolvem, por exemplo, modelos para descrição de objetos de aprendizagem, padrões de interoperabilidade e linguagens de modelagem (BARKER, 2005; FRIESEN, 2005; VIEIRA; COSTA; RAABE, 2012). Essas abordagens podem ser utilizadas de forma individualizada ou combinadas entre si.

Com relação aos modelos para descrição dos dados dos objetos de aprendizagem destaca-se o padrão IEEE LOM (*Learning Object Metadata*). Através dele, é possível especificar quais aspectos de um objeto de aprendizagem deveriam ser descritos e quais os vocabulários a serem usados para essas descrições (BARKER, 2005). O modelo LOM é a base de outra especificação de modelos de dados de objetos de aprendizagem: a IMS *Learning Resource Metadata* (IMS GLOBAL LEARNING CONSORTIUM, 2003). Outro modelo também utilizado para representação dos metadados de objetos de aprendizagem é o *Dublin-Core Metadata* (FRIESEN, 2005).

Para possibilitar a interoperabilidade de objetos de aprendizagem entre ambientes virtuais de aprendizagem, existem modelos como o SCORM (Shareable Content Object Reference Model) e o IMS *Common Cartridge* (VIEIRA; COSTA; RAABE, 2012). Essas duas especificações são compostas por modelos que permitem, além da interoperabilidade, a publicação, distribuição, acessibilidade e reutilização de conteúdos digitais entre diferentes ambientes de aprendizagem.

Com o objetivo de obter um modelo mais completo que fosse além da representação de objetos de aprendizagem, o IMS *Global Learning Consortium* (IMS GLOBAL LEARNING CONSORTIUM, 2003) criou a especificação IMS *Learning Design* (IMS LD). Segundo Qu e He (2009), as principais vantagens dele com relação a outras especificações são: enquanto o SCORM é baseado em um modelo de aprendizagem onde um único aluno interage com o sistema, o IMS LD permite a especificação de situações com vários alunos interagindo entre si e com o professor em ações individuais ou colaborativas; o IMS LD avança da visão de objetos de aprendizagem isolados para considerar as atividades de aprendizagem que acontecem no ambiente virtual, as quais incluem pessoas, sequenciamento de ações, ferramentas e serviços; e, ao considerar atividades de aprendizagem executadas por pessoas com diferentes papéis (como aluno iniciante e aluno avançado), o IMS LD permite a especificação de roteiros de aprendizagem personalizados.

2.5.1 A especificação IMS Learning Design

A especificação IMS *Learning Design* é um modelo aberto usado para representar uma grande variedade de estratégias pedagógicas de maneira formal, semântica, interoperável e legível por computador (KOPER; MIAO, 2007). Segundo Derntl *et al.* (2012), o modelo IMS LD é o único que permite a definição e orquestração de forma integrada de fluxos de atividades, recursos (ferramentas e serviços) e participantes com diferentes papéis.

O consórcio responsável pela criação da especificação IMS LD, *IMS Global Learning Consortium*, é composto por dezenas de instituições educacionais, universidades e empresas de tecnologia, tais como: Pearson Education, Australian Government Department of Education, New York Department of Education, Fundação Getúlio Vargas, California State University, Open University Japan, Open University Netherlands, Seoul Cyber University, Universitat Oberta de Catalunya, Microsoft, Oracle, Samsung Electronics, entre outras. O principal objetivo do *IMS Global Learning Consortium* é promover avanços inovadores na educação através da criação de padrões abertos para conteúdos e aplicações educacionais.

Derntl *et al.* (2012) fizeram um estudo com 40 professores sobre o uso do modelo IMS LD para representação de atividades de aprendizagem. A principal conclusão deles indica que a estrutura conceitual do IMS LD foi bem aceita pelos professores e não apresentou grandes desafios para sua compreensão: mais de 75% dos participantes conseguiu desenvolver modelos em conformidade com o modelo.

Oliveira, Salvador e Novais (2014) classificam a especificação IMS LD como uma ontologia para modelagem conceitual de projetos educacionais. Eles realizaram um estudo com 29 alunos do curso de ciência da computação e concluíram que o uso dessa ontologia pode proporcionar a construção de softwares educativos com menos erros.

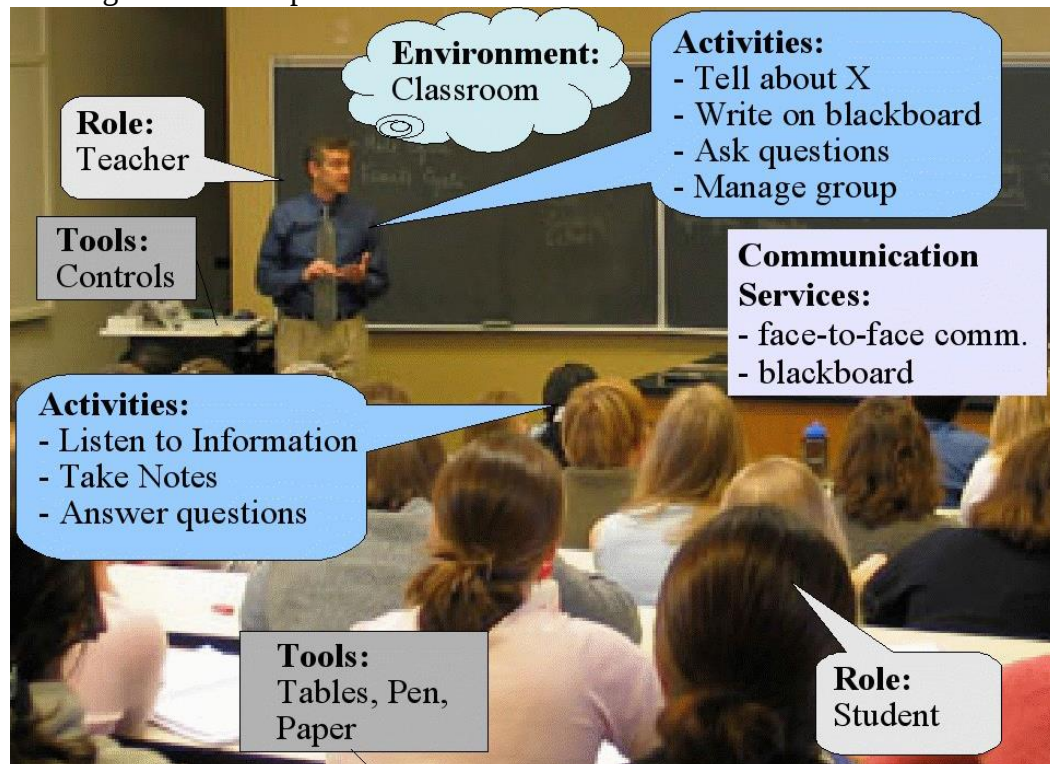
É importante esclarecer que o conceito de *Learning Design* é mais amplo que o modelo IMS LD. *Learning Design* refere-se à representação dos processos de ensino e aprendizagem (BRITAIN, 2004). É uma definição bastante abrangente, que não é recente e pode ter como exemplo os tradicionais planos de aula. Ao considerar a utilização das tecnologias na educação, através do *Learning Design*, é possível representar novos recursos digitais para incrementar a qualidade dos processos de ensino e aprendizagem. Nesse contexto, o objetivo da especificação *IMS Learning Design* é fornecer um modelo computacional para descrever estruturas de atividades de aprendizagem e seus relacionamentos com os participantes e os recursos envolvidos em práticas educativas suportadas por tecnologias digitais.

A Figura 9 mostra os principais conceitos do IMS LD. Uma pessoa pode assumir diferentes papéis (*<roles>*), como professor ou aluno. Esses papéis podem executar diferentes atividades (*<activities>*). O ambiente (*<environment>*), nesse exemplo a sala de aula, consiste do local onde acontece a prática educativa, que pode conter ferramentas (*<tools>*) e serviços (*<communication services>*).

A imagem utilizada é um objeto estático. Entretanto, o processo de ensino e aprendizagem é dinâmico. Assim, não é possível através dela demonstrar todos os elementos da IMS LD, como aqueles relacionados à descrição dos processos envolvidos. As próximas

seções apresentam a estrutura do IMS LD, exemplos de ferramentas compatíveis e algumas dificuldades encontradas para ampliação do seu uso.

Figura 9 – Exemplo com os elementos da IMS LD em uma sala de aula



Fonte: Koper e Miao (2007).

2.5.2 A estrutura da linguagem do IMS LD

O modelo IMS LD é composto por uma linguagem específica de domínio (DSL) e outras especificações diferentes para representar processos de ensino e aprendizagem, dentre as quais: *IMS Resource Meta-Data Specification* (IMS RMS), para padronização dos metadados dos recursos digitais; *IMS Content Packing* (IMS CP), para empacotamento e distribuição; *IMS Simple Sequencing* (IMS SS), para sequenciamento das atividades de aprendizagem.

A linguagem utilizada é baseada na especificação *Educational Modelling Language* (EML) criada pela Universidade Aberta da Holanda (TATTERSALL; KOPER, 2003). A EML tinha como objetivo principal ser uma meta-linguagem que suportasse diversidade pedagógica e inovação, bem como promovesse a troca e a interoperabilidade de materiais de *e-learning* (IMS LD, 2003). Essa diversidade pedagógica consiste em permitir que as atividades de aprendizagens modeladas através da EML variem de acordo com três fundamentos: atividades individuais ou em grupo de alunos; se em grupo, podem ser

independentes ou colaborativas; e, podem seguir uma abordagem behaviorista, cognitivista, construtivista ou outra linha pedagógica.

Através da EML, é possível representar um processo de ensino e aprendizagem da seguinte forma: métodos estabelecem a sequência em que atividades são executadas por alunos e professores e cada atividade contém objetos e serviços que compõem um ambiente. Para dar suporte a essa representação, ainda existem outros elementos importantes: propriedades, condições e notificações. Todos esses elementos, quando integrados para representar a prática educativa, constituem uma Unidade de Aprendizagem (em inglês *Unit of Learning* – UoL).

A modelagem através da especificação IMS LD é dividida em três níveis, que são:

- Nível A: Inclui os principais elementos da linguagem, tais como atividades, papéis, métodos e ambiente. Assim, o nível A contém o núcleo da especificação necessário para representar as unidades de aprendizagem. Os níveis B e C são complementares.
- Nível B: Adiciona as propriedades e as condições ao nível A. Através delas, é possível a personalização e elaboração de sequenciamentos e interações baseadas em verificações.
- Nível C: Adiciona notificações ao nível B, mais especificamente às propriedades, tornando possível a mudança de seus valores após a ocorrência de eventos.

A especificação IMS LD, além de possibilitar a modelagem de práticas educativas de uma forma geral, atende aos seguintes requisitos:

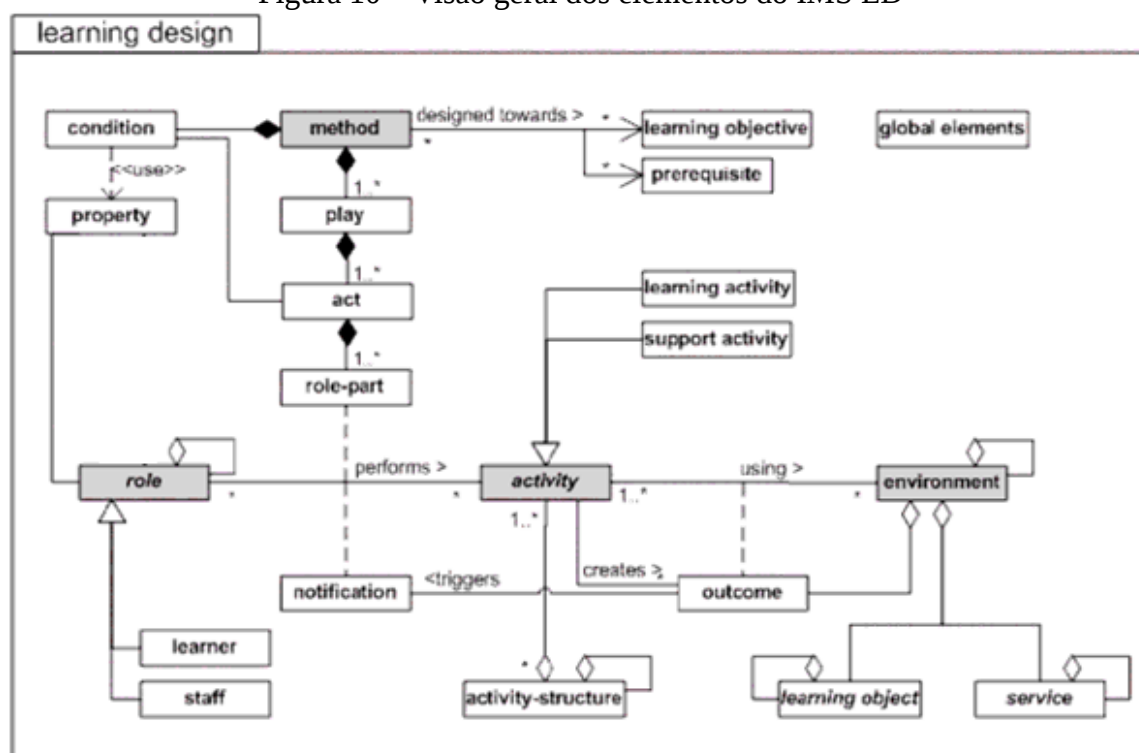
1. Plenitude. A modelagem deve ser capaz de representar todos os elementos envolvidos em uma unidade de aprendizagem, inclusive a referência a objetos de aprendizagem e serviços.
2. Flexibilidade pedagógica. A descrição deve ser flexível de forma a não se restringir a uma determinada abordagem pedagógica.
3. Personalização. A especificação deve ser capaz de descrever unidades de aprendizagem que contemplem conteúdos e atividades que possam ser adaptados de acordo com determinadas condições, como preferências, necessidades educacionais ou circunstâncias em que se encontram os alunos.
4. Formalização. A linguagem deve descrever a unidade de aprendizagem de forma que essa possa ser processada automaticamente.
5. Reprodutibilidade. A especificação deve descrever a unidade de aprendizagem de tal forma que possa ser possível executá-la repetidas vezes em diferentes configurações e com diferentes pessoas.

6. Interoperabilidade. Os modelos das unidades de aprendizagem devem poder ser compreendidos por diferentes sistemas.
7. Compatibilidade. A especificação deve utilizar padrões reconhecidos, tais como IMS CP, IMS RMS e IMS SS.
8. Reuso. Os modelos gerados devem poder ser reutilizados em outras situações.

O modelo conceitual da IMS LD é expresso como um conjunto de diagramas de classes e os respectivos vocabulários usados. O detalhamento completo de todos os elementos pode ser obtido através da especificação completa (IMS LD, 2003). A Figura 10 apresenta uma visão geral do modelo, com destaque para os principais elementos (em cinza) descritos a seguir:

- *activity*: representa a atividade de aprendizagem (*learning activity*), mas que também pode ser uma atividade de suporte (*support activity*) realizada por um professor.
- *role*: representa os papéis que podem ser executados pelos participantes durante a prática educativa, podendo ser do tipo aluno (*learner*) ou professor (*staff*).
- *method*: representa como será o sequenciamento das atividades na unidade de aprendizagem através dos elementos *play* e *act*.
- *environment*: representa os objetos de aprendizagem (*learning object*) e serviços (*service*) que podem ser utilizados em uma atividade de aprendizagem (*learning activity*).

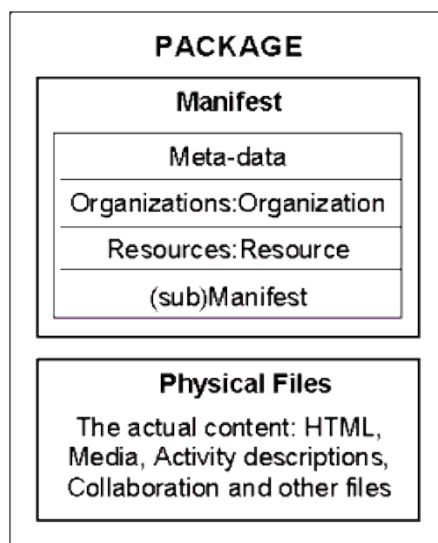
Figura 10 – Visão geral dos elementos do IMS LD



Fonte: IMS LD (2003).

Após a modelagem utilizando a especificação IMS LD, o resultado final é um pacote seguindo o IMS CP contendo a unidade de aprendizagem. A estrutura do IMS *Content Packing* é apresentada na Figura 11, a qual é composta por: um arquivo com o manifesto (*Manifest*) que contém a especificação da unidade de aprendizagem em formato XML, com nome pré-definido (imsmanifest.xml); e outros arquivos (*Physical Files*) que são necessários para execução daquela UoL, como imagens, vídeos e arquivos HTML. O uso dessa estrutura (arquivo XML e arquivos de conteúdo) faz com que os projetos modelados em IMS LD possam ser lidos por sistemas computacionais (modelos legíveis por computador).

Figura 11 – Estrutura do IMS Content Packing

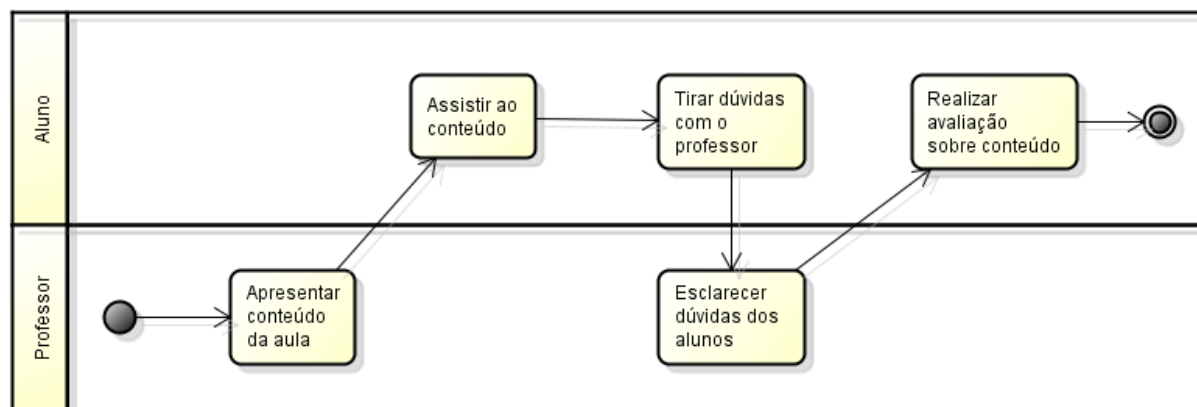


Fonte: IMS LD (2003).

Para ilustrar um exemplo de especificação em IMS LD, a

Figura 12 apresenta um Diagrama de Atividades da UML com ações que mostram interações entre aluno e professor em uma aula presencial. Nesse exemplo, o professor apresenta o conteúdo da aula, esclarece as dúvidas dos alunos e, por fim, faz uma avaliação sobre o conteúdo lecionado. A Figura 13 e a Figura 14 mostram como ficaria a especificação em IMS LD dessa prática educativa, apenas nas partes referentes aos papéis e às atividades.

Figura 12 – Diagrama de atividades com exemplo de aula presencial.



Fonte: elaborada pelo autor.

A partir da Figura 13, observa-se os dois elementos `<roles>`, sendo um do tipo `<learner>` e outro do tipo `<staff>`. Também se percebe o atributo e o elemento em comum nesses dois tipos de papel: *identifier*, que funciona como uma chave-primária, e o *<title>*, que indica o título daquele papel. O papel `<learner>` tem um atributo diferente para indicar que, em tempo de execução, podem ser criados novos elementos desse tipo: *create-new*.

Figura 13 – Exemplo de codificação em IMS LD do elemento `<role>`

```

<imsld:roles>
  <imsld:learner identifier="role-0302e389" create-new="allowed" >
    <imsld:title>Aluno</imsld:title>
  </imsld:learner>
  <imsld:staff identifier="role-57be18a1">
    <imsld:title>Professor</imsld:title>
  </imsld:staff>
</imsld:roles>

```

Fonte: elaborada pelo autor.

A Figura 14 mostra trecho de código que equivale à especificação da atividade “Apresentar conteúdo da aula”. Cada uma das outras quatro atividades tem codificação semelhante. Pode-se observar que uma `<learning-activity>` conta com os seguintes atributos e elementos: *identifier*, identificador único; *isvisible*, para indicar se a atividade deve ser visível; *<title>*, título da atividade; *activity-description*, descrição da atividade; *<item>*, que referencia um arquivo que tem a descrição da atividade; e, *<complete-activity>*, que indica como a atividade deve ser completada.

Figura 14 – Exemplo de codificação em IMS LD do elemento <learning-activity>.

```
<imsld:activities>
  <imsld:learning-activity identifier="la-d1d662e2" isvisible="true">
    <imsld:title>Apresentar conteúdo da aula</imsld:title>
    <imsld:activity-description>
      <imsld:item identifier="fb2abf79" identifierref="fb2abf79.txt"
    </imsld:item>
    </imsld:activity-description>
    <imsld:complete-activity>
      <imsld:user-choice />
    </imsld:complete-activity>
  </imsld:learning-activity>
```

Fonte: elaborada pelo autor.

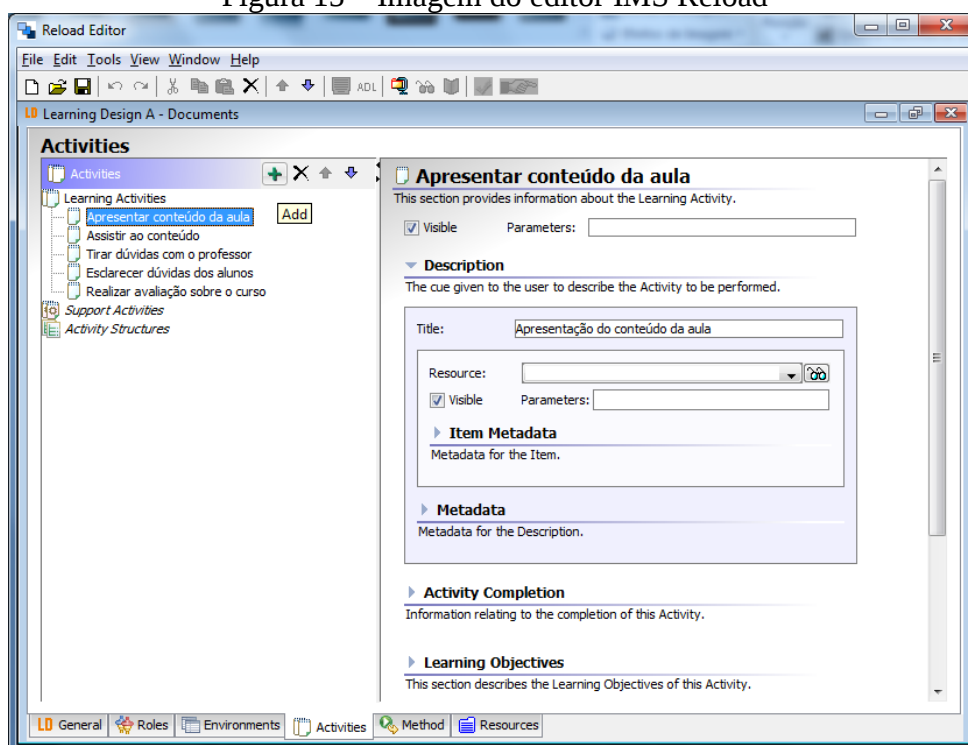
2.5.3 Ferramentas compatíveis com a especificação IMS LD

Conforme se pode perceber a partir dos trechos de códigos ilustrados, a especificação de uma prática educativa em IMS LD, por mais simples que sejam as atividades, torna-se complexa e extensa. Portanto, torna-se necessária a utilização de ferramentas de modelagem para auxiliar a especificação das unidades de aprendizagem.

Um dos primeiros softwares utilizados para edição de práticas educativas com IMS LD foi o Reload³. Além de permitir a construção de unidades de aprendizagem no formato IMS *Learning Design*, o Reload também dá suporte aos seguintes padrões: IMS *Resource Meta-Data Specification*, IMS *Content Packing*, SCORM 1.2 e SCORM 2004. Conforme pode ser observado na Figura 15, o Reload é um editor que permite ao usuário adicionar os elementos, que ficam do lado esquerdo da tela, e preencher os atributos com os respectivos valores, do lado direito.

³ <http://www.reload.ac.uk/>

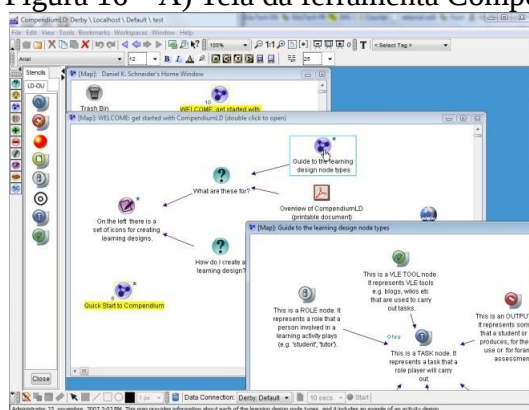
Figura 15 – Imagem do editor IMS Reload



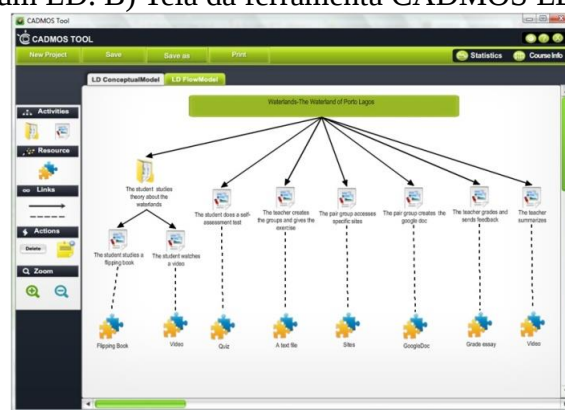
Fonte: capturada pelo autor.

Com o objetivo de facilitar a construção de unidades de aprendizagem, diferentes pesquisadores desenvolveram ferramentas que possibilitam a modelagem gráfica dos elementos da IMS LD. A Figura 16 ilustra duas dessas ferramentas: Compendium LD e CADMOS LD.

Figura 16 – A) Tela da ferramenta Compendium LD. B) Tela da ferramenta CADMOS LD



A)



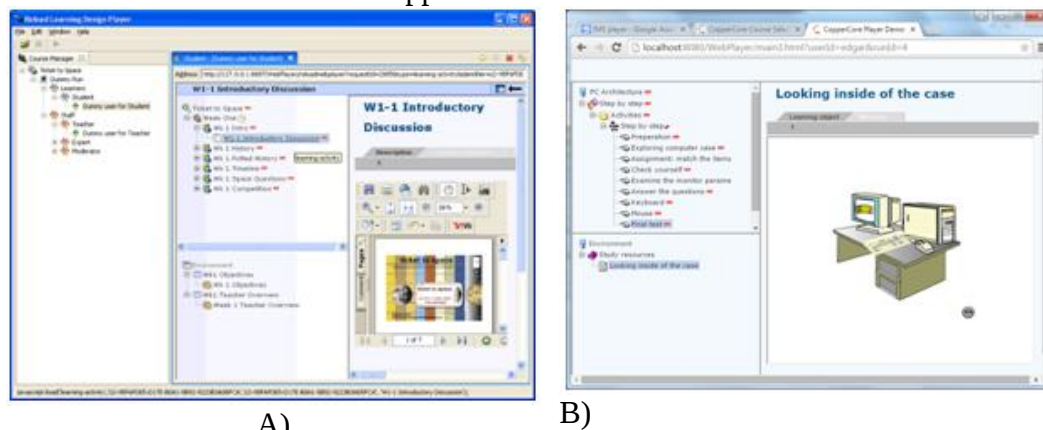
B)

Fonte: capturadas pelo autor.

De maneira semelhante à edição, para interpretação e execução de uma unidade de aprendizagem especificada em IMS LD existem ferramentas apropriadas. Conhecidas como *players*, podem ser aplicações independentes ou módulos específicos nos próprios ambientes virtuais de aprendizagem. A Figura 17 ilustra dois exemplos: a aplicação desktop Reload LD

Player; e o servidor web CopperCore mostrando uma página web resultado da execução de uma UoL em IMS LD.

Figura 17 – A) Tela da ferramenta Reload Player B) Página web com a execução de uma UoL em IMS LD via CopperCore



Fonte: capturadas pelo autor.

2.5.4 Dificuldades para o uso do IMS LD

Apesar de apresentar uma estrutura que possibilita a modelagem de processos com diferentes abordagens pedagógicas, contar com ferramentas para edição e execução, e ter vários casos de sucesso de seu uso na prática, existem alguns entraves que impedem uma maior utilização do modelo IMS LD. Além disso, como afirmado por Moura Filho (2007), o IMS LD não é a única opção para representação de práticas educativas.

Burgos (2010) realizou um estudo detalhado para identificar os principais pontos fracos da especificação IMS LD, tomando como base cinco diferentes unidades de aprendizagem. Os maiores problemas encontrados foram: a definição e utilização das propriedades é uma tarefa difícil; a estrutura para definição das condições torna o processo de edição complicado; não existe forma de configuração de conexão a bases de dados diferentes; não há forma de interação com ferramentas externas no sentido de troca de informações; e a IMS LD não prevê mecanismos para armazenamento de blocos de dados.

A partir das deficiências encontradas, Burgos (2010) propõe um conjunto de alterações e extensões ao IMS LD, dentre as quais destacamos: ampliação das estruturas de controle com os novos elementos *<case>*, *<for>* e *<while>*; criação do novo elemento *<interaction>* para possibilitar as interações entre as atividades; adição do elemento *<goto>* para direcionar o fluxo de aprendizagem; acréscimo dos elementos *<export>*, *<import>*, *<to-db>* e *<from-db>* para permitir o registro e consulta das informações seja em arquivos ou em bases de dados.

Já no estudo apresentado por Derntl *et al.* (2012), os professores apontaram as seguintes dificuldades acerca do uso do IMS LD para modelagem de unidades de aprendizagem: muitos não souberam fazer a associação dos participantes (*<roles>*) com as atividades (*<learning-activities>*) através do elemento *<role-part>*; alguns também tiveram problemas no gerenciamento do fluxo das informações entre as atividades; por fim, os participantes relataram que as condições (*<conditions>*) eram os elementos mais complicados, porque eles não sabiam quais outros elementos elas controlavam e em que momentos elas eram avaliadas.

2.5.5 A IMS LD e as aulas de campo ubíquas

No caso das aulas de campo, um complicador à utilização da especificação IMS LD na forma original é a ausência de conexão com a Internet em diversos momentos no campo. Esse fato dificulta o uso da IMS LD porque seus elementos foram desenvolvidos considerando alunos conectados ao ambiente virtual de aprendizagem. Por exemplo, no caso dos serviços de comunicação entre os participantes, a IMS LD prevê apenas os elementos *<conference>* e *<send-mail>*, que não contemplam a especificação de serviços como a troca de arquivos e envio de mensagens utilizando tecnologias sem fio.

Outra limitação da IMS LD, com relação a sistemas ubíquos, diz respeito à ausência nesse modelo de elementos para especificação da captura de dados dos sensores dos dispositivos móveis, como câmera, microfone, GPS e acelerômetro. Um elemento que poderia ser utilizado para contornar essa limitação seria o *<learning-object>*. Entretanto, conforme De-la-Fuente-Valentín, Pardo e Kloos (2011), apesar de ampliar os recursos da UoL, o *<learning-object>* não possibilita a integração da unidade de aprendizagem com ferramentas externas.

Além das limitações exemplificadas, outros requisitos importantes para sistemas ubíquos voltados a aulas de campo não foram previstos quando a IMS LD foi elaborada, porque o foco da especificação eram os sistemas educativos online via Internet. Por exemplo, a persistência local de dados não é prevista pela IMS LD. Essa é uma funcionalidade necessária para as aulas de campo tendo em vista a não garantia de conexão com a Internet nos locais onde elas são realizadas e a indispensabilidade de salvar as informações registradas pelos alunos. Outro exemplo é o suporte ao uso de informações contextuais em campo, que não é contemplado de forma direta pela especificação IMS LD. Esse é um requisito importante, pois em diversos momentos da aula de campo, surgem situações onde ações específicas devem ser executadas de acordo com o contexto do aluno, como a localização em

que ele se encontra ou as atividades que ele já executou naquela aula. Assim, percebe-se que extensões e adaptações são necessárias para possibilitar a modelagem desse tipo de sistema usando o modelo IMS LD.

2.6 Conclusão

Este capítulo apresentou os principais assuntos que embasam teoricamente este trabalho e que são de fundamental importância para o desenvolvimento do ambiente proposto nesta tese de doutorado. Inicialmente foram apresentados os conceitos essenciais relacionados com o desenvolvimento de sistemas ubíquos e, posteriormente, a abordagem do desenvolvimento dirigido por modelos foi descrita. Considerando a área de aplicação deste trabalho, em seguida, foram apresentadas definições e características importantes associadas ao uso dos sistemas ubíquos na educação. Por fim, conceitos da área de modelagem de processos e recursos educacionais foram introduzidos e a especificação IMS *Learning Design* foi detalhada. O próximo capítulo apresenta um levantamento de trabalhos relacionados ao ambiente apresentado nesta tese de doutorado.

3 TRABALHOS RELACIONADOS

Neste capítulo são descritos trabalhos importantes que apresentam soluções relacionadas à modelagem e geração de sistemas ubíquos voltados especificamente para aulas de campo ou que podem ser aplicados nesse tipo de prática educativa. Nas seções 3.1 a 3.5, os sistemas são detalhados e, na seção 3.6, uma análise comparativa entre os trabalhos relacionados é apresentada.

3.1 Lemonade (GIEMZA; BOLLEN; HOPPE, 2011)

Giemza, Bollen e Hoppe (2011) apresentam um sistema que possibilita aos professores projetarem aulas de campo e exportarem aplicações para os alunos utilizarem em *smartphones*: o LEMONADE (*“Learning Environment for Mobile Network-Able Devices”*). O sistema permite a formação de grupos de alunos e a definição de dicas e atividades para serem apresentadas aos participantes de acordo com a localização deles durante a aula de campo. A Figura 18 mostra a tela principal do sistema LEMONADE. O sistema LEMONADE funciona como um plug-in no ambiente de modelagem gráfica FreeStyler (HOPPE; GAßNER, 2002), para possibilitar a edição e planejamento das aulas de campo.

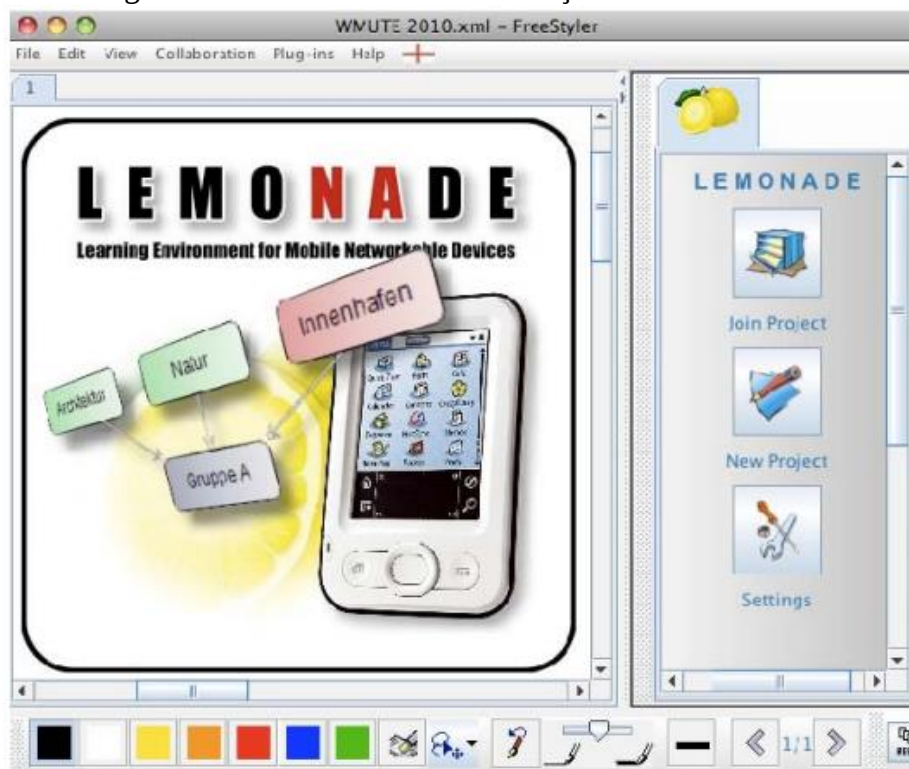
Através do LEMONADE, o professor pode construir um projeto de aula de campo composto por uma sequência de tarefas as quais o aluno deverá executar, que podem ser: anotação de textos, captura de áudios (via microfone) e fotos (via câmera) e obtenção da localização (via GPS). Após o projeto da aula ter sido elaborado, o aluno pode fazer o download da aplicação para o seu *smartphone* Windows Mobile. O aplicativo gerado não necessita de acesso a Internet durante a aula de campo. Quando retornam para sala de aula, os dados coletados podem ser atualizados no sistema LEMONADE.

Por gerar aplicações apenas na linguagem C#, o LEMONADE restringe seu uso e impossibilita sua utilização em dispositivos móveis com sistemas operacionais largamente utilizados como Android e iOS. A ferramenta também é limitada em termos de recursos, não permitindo ao aluno, por exemplo: gravar vídeos ou utilizar outros sensores além do GPS, como acelerômetro e bússola, que são importantes em aulas de campo.

A estrutura de uma aula projetada com o LEMONADE trata-se de um fluxo sequencial linear que não permite a execução de aulas adaptativas. Portanto, não é possível, por exemplo, construir uma aula que varie de acordo com o contexto do aluno. A ferramenta não conta com

elementos para especificação de condições que modifiquem o fluxo da aprendizagem para alunos em situações diferentes.

Figura 18 – Tela do ambiente de edição do LEMONADE



Fonte: Giemza, Bollen e Hoppe (2011).

3.2 UoLmP (GÓMEZ, 2013)

A tese de doutorado de Gómez (2013) apresenta uma abordagem para especificação de práticas educativas adaptativas e sensíveis ao contexto para dispositivos móveis baseada no modelo IMS LD. Desde 2010, Gómez tem publicado, com outros autores, trabalhos sobre adaptação de conteúdo com sensibilidade ao contexto para dispositivos móveis (GÓMEZ; FABREGAT, 2010; GÓMEZ *et al.*, 2014).

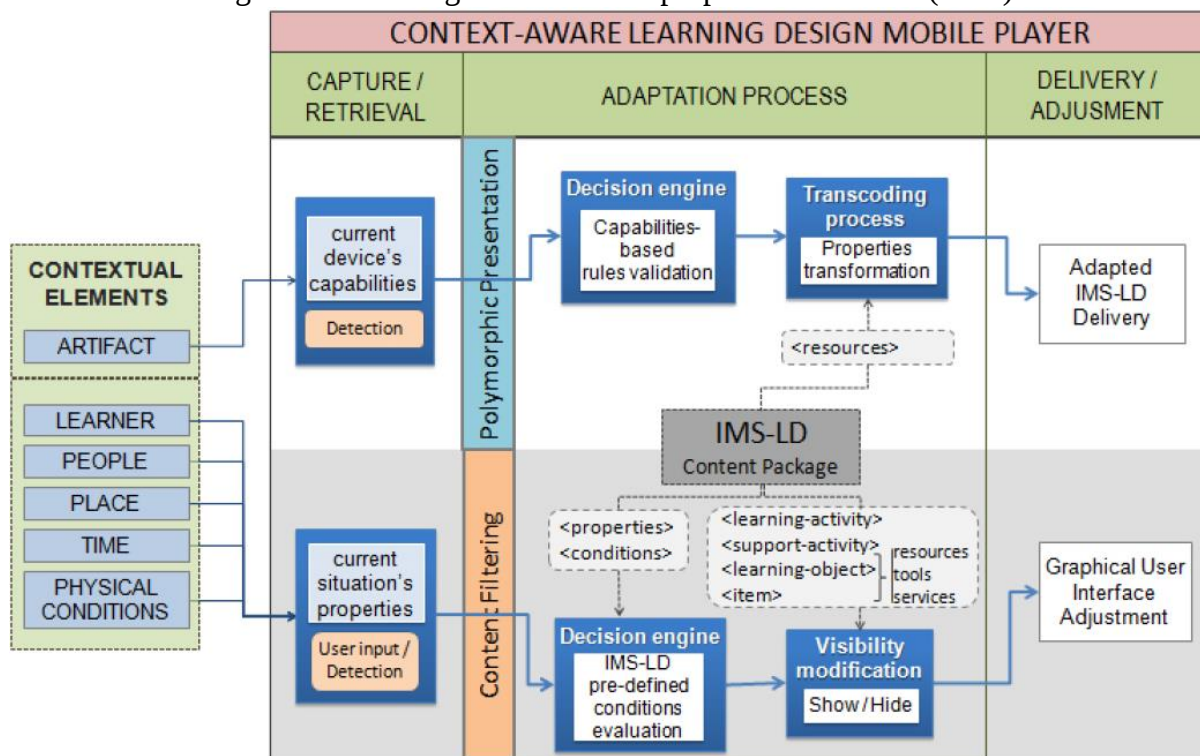
Um dos principais resultados apresentados na tese de Gómez (2013) é o UoLmP (*Unit of Learning mobile Player*). Este consiste em um software para a plataforma Android capaz de interpretar unidades de aprendizagem escritas em IMS LD e adaptar-se dinamicamente de acordo com as mudanças de contexto em tempo real.

O UoLmP funciona da seguinte forma: existe um repositório de projetos de Unidades de Aprendizagem (UoL) disponível em um servidor; quando o aluno solicita o download de uma dessas UoL, o UoLmP inicia o processo de adaptação pela avaliação das características do dispositivo móvel (e.g. resolução do display e suporte à execução de vídeos); então, o

UoLmP faz o download de uma UoL mais adequada às características daquele dispositivo móvel; o UoLmP captura automaticamente as informações contextuais através dos sensores de acordo com um modelo de contexto pré-definido, armazenando-as em elementos *<properties>* da IMS LD; ao logo da execução daquela UoL no dispositivo móvel, o UoLmP utiliza os valores das *<properties>* para realizar as adaptações de conteúdo.

A Figura 19 mostra uma visão geral da tese do Gómez (2013). Através dessa imagem pode-se observar: os elementos contextuais utilizados; o processo de adaptação, que é responsável por criar e armazenar as configurações necessárias para as adaptações em tempo real e as diferentes UoL; e o processo da entrega do conteúdo, que é realizado pelo UoLmP no dispositivo móvel do aluno.

Figura 19 – Visão geral do modelo proposto em Gómez (2013)



Fonte: Gomez, 2013.

Considerando a utilização prática por professores e projetistas de cursos online, a principal deficiência da proposta de Gómez (2013) é a ausência de uma ferramenta de autoria. Inclusive, essa questão é apontada por ele como uma possibilidade de trabalho futuro. Na tese, o autor propõe a utilização de editores IMS LD existentes.

Entretanto, a não existência dessa ferramenta pode comprometer a utilização de todo o restante da proposta, pois erros na especificação da UoL poderão inviabilizar a estrutura apresentada. É importante reforçar que a dificuldade para edição de unidades de

aprendizagem em IMS LD é apontada em diferentes trabalhos como um dos pontos fracos desse modelo (BURGOS, 2010; MARTÍNEZ-ORTIZ; SIERRA; FERNÁNDEZ-MANJÓN, 2009).

Outras duas deficiências estão mais relacionadas à questão da mobilidade dos dispositivos. A abordagem de Gómez (2013) assume que para a realização da adaptação existe a necessidade do aplicativo se conectar a um servidor para obter informações, porém em determinadas situações onde não há conexão (e.g. aulas de campo) essa adaptação pode se tornar inviável. Outra questão, apontada no próprio trabalho de Gómez (2013), é a impossibilidade de armazenar e utilizar bancos de dados. Esta é uma limitação da própria IMS LD, que pode dificultar seu uso em aplicações ubíquas que precisam gerenciar informações localmente no dispositivo móvel.

3.3 MAT for ARLearn (TABUENCA *et al.*, 2014)

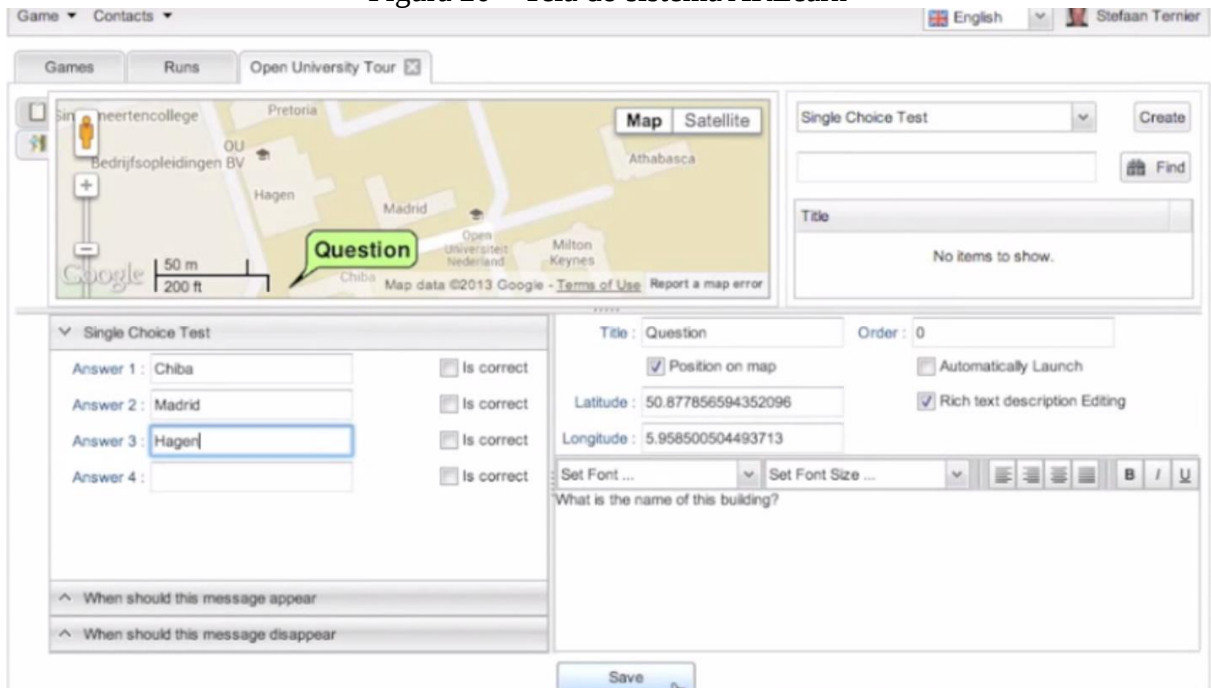
Tabuenca *et al.* (2014) apresentam uma ferramenta de autoria de aplicações móveis educativas: *Mobile Authoring Tool for ARLearn* (MAT for ARLearn). Esta ferramenta foi construída sobre o ARLearn, o qual consiste em um sistema para educadores criarem, gerarem e executarem jogos sérios⁴ em dispositivos móveis na plataforma Android.

O sistema ARLearn (Figura 20) conta com uma aplicação Web no servidor para edição dos jogos e um aplicativo Android que executa o jogo gerado no dispositivo móvel (TERNIER *et al.*, 2012). Dentre os possíveis cenários de jogos sérios construídos com o ARLearn estão as aulas de campo. Os jogos projetados com o ARLearn ficam armazenados na plataforma de aplicações *Google App Engine* (GAE) para serem acessados pelo cliente no dispositivo móvel.

O MAT for ARLearn (Figura 21) consiste em um software que permite a construção dos jogos sérios no próprio dispositivo móvel do usuário (com sistema operacional Android). A principal contribuição dele para o ARLearn é a possibilidade de projetar novos ou personalizar aplicativos existentes utilizando dados capturados (e.g. fotos, áudios e vídeos) no mesmo ambiente nos quais eles serão jogados. Assim, ele proporciona uma autoria contextualizada.

⁴ Jogos sérios são jogos cujo objetivo, desde a fase de concepção do jogo, vai além do entretenimento puro. Jogos educacionais e exergames são exemplos de jogos sérios (SÁNCHEZ; OLIVARES, 2011).

Figura 20 – Tela do sistema ARLearn



Fonte: capturada pelo próprio autor.

O MAT for ARLearn e o ARLearn mantêm a mesma estrutura e as aplicações móveis projetadas apresentam as mesmas características. Entretanto, estas são restritas ao sistema operacional Android e são executadas através do aplicativo cliente do ARLearn, que necessita de conexão com a Internet para executá-las. Além disso, a única informação contextual suportada é a localização, que pode ser obtida via GPS, RFID ou QRCode. Por fim, o sistema não utiliza modelos padrões para representação dos processos de ensino e aprendizagem, o que dificulta a interoperabilidade com outras ferramentas.

Figura 21 – Telas do sistema MAT for ARLearn



Fonte: Tabuenca *et al.*, 2014.

3.4 Midgar (GARCÍA *et al.*, 2014)

A Internet das Coisas (em inglês *Internet of Things*, IoT) é um paradigma na área das tecnologias sem fio que tem como ideia principal a onipresença em um ambiente de diversos objetos tecnológicos (como sensores, *smartphones*, etiquetas RFID, entre outros) que estão interconectados, interagindo e cooperando entre si para alcançar determinados objetivos (ATZORI; IERA; MORABITO, 2010). Dentre esses, está o uso para favorecer os processos de ensino e aprendizagem. Considerando o cenário de aplicações com fins educativos, as soluções em *Internet of Things* e *Ubiquitous learning* se confundem.

Nesse contexto, o trabalho de García *et al.* (2014) apresenta um ambiente IoT para conectar diferentes objetos ubíquos e heterogêneos, criados a partir de uma ferramenta gráfica que gera modelos em uma DSL própria. O principal objetivo dos autores é permitir que autores sem conhecimentos de programação construam objetos para serem usados em seus domínios de conhecimento com o suporte das tecnologias IoT.

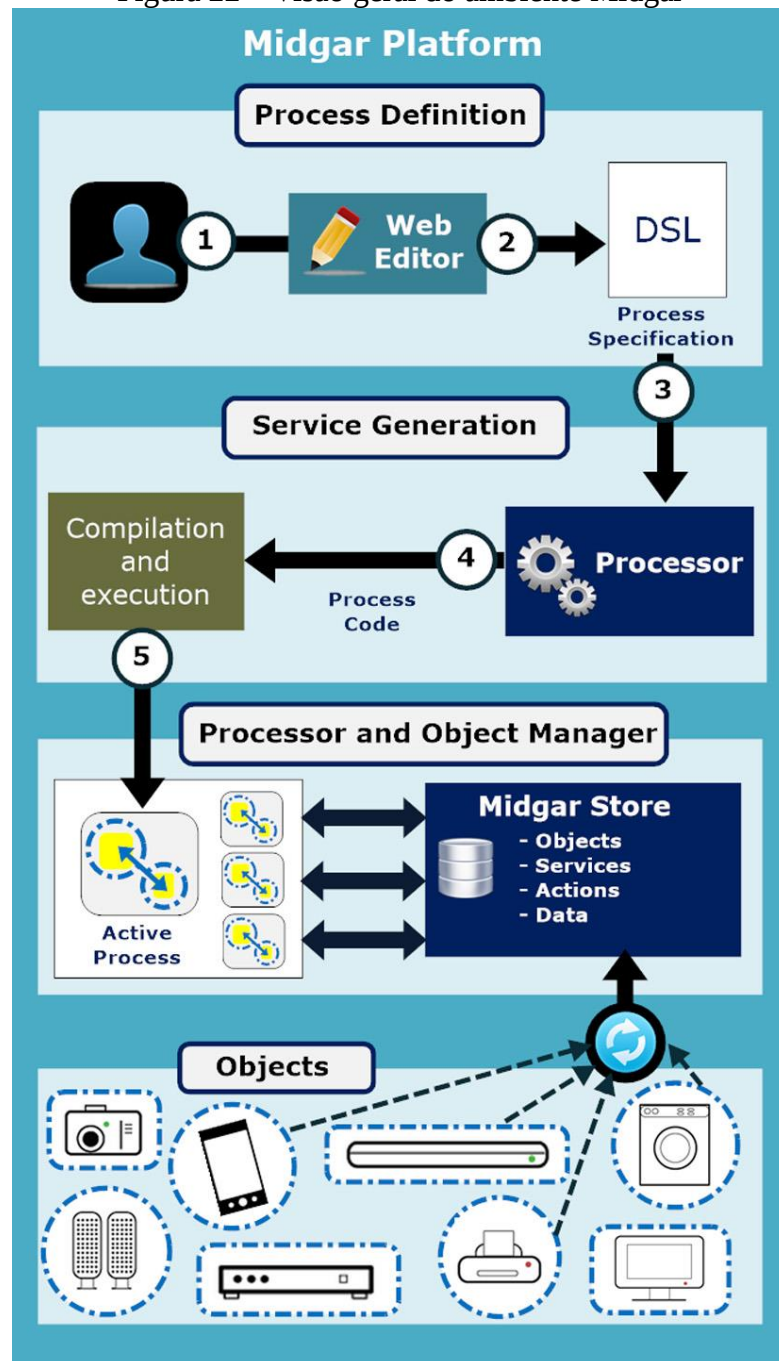
A arquitetura do ambiente Midgar pode ser dividida em quatro camadas, conforme apresentado na Figura 22: *Process Definition*, onde o usuário deve projetar sua aplicação através da ferramenta de edição, a qual gerará uma especificação em XML na DSL do Midgar; *Service Generation*, que processa o arquivo XML gerado pela ferramenta de edição e gera uma aplicação Java; *Processor and Object Manager*, que compila o código fonte gerado em Java para Android e disponibiliza a aplicação no servidor; e *Objects*, que mantém o registro dos serviços dos *Smart Objects* disponíveis para interação no ambiente. Um *Smart Object* é um objeto físico ou digital com recursos de sensoramento, processamento e conexão à rede (KORTUEM *et al.*, 2010).

As aplicações que podem ser projetadas pelos usuários no Midgar consistem essencialmente em chamadas de ações dos *Smart Objects* de acordo com determinadas condições. Por exemplo, é possível projetar uma aplicação para *smartphone* que realiza a seguinte execução: acessa um “*smart*” termômetro, recupera a temperatura do ambiente e, no caso de estar quente, envia uma mensagem para um “*smart*” ar-condicionado informando que ele deve esfriar mais o local. Assim, o ambiente caracteriza-se como um sistema de propósito geral que permite o projeto e execução em dispositivos móveis de aplicações que interconectam *smart objects*, focadas no que eles podem capturar e executar.

Dessa forma, o Midgar não contempla funcionalidades importantes para processos de ensino e aprendizagem suportados por tecnologias ubíquas, tais como: interação entre os participantes das práticas educativas (alunos e professor); inserção de recursos digitais de

aprendizagem nas aplicações (como vídeos, textos e imagens); e execução de forma diferenciada da aplicação de acordo com o perfil do usuário. Além disso, o Midgar depende de conexão com a Internet para chamar os serviços dos *smart objects*, e em algumas situações não há conectividade durante as práticas educativas.

Figura 22 – Visão geral do ambiente Midgar



Fonte: García et al. (2014).

3.5 App Inventor (MIT, 2012)

O App Inventor é um sistema para construção de aplicações genéricas para serem executadas em *smartphones* e tablets que utilizem o sistema operacional Android. O App Inventor foi criado em 2010 pela empresa Google. Desde 2012, o App Inventor é mantido pelo Massachusetts *Institute of Technology* (MIT) dos Estados Unidos. A Figura 23 mostra a tela principal do sistema App Inventor para construção dos aplicativos. Pode-se observar a tela principal com um retângulo central que representa a tela do celular e dos lados esquerdo e direito os recursos que podem ser utilizados pelo usuário da ferramenta.

Conforme Turbak e Okerlund (2013), o App Inventor é uma ferramenta amplamente utilizada, contando com mais de 880 mil usuários que desenvolveram mais de dois milhões de aplicativos. A área educacional concentra a maioria dos casos de sucesso das aplicações geradas pelo App Inventor.

Entre os exemplos de aplicações educacionais geradas pelo App Inventor podemos citar: o “Plant Data Collector”, um aplicativo desenvolvido por uma professora de biologia da Clayton State University nos Estados Unidos para auxiliar as aulas de campo de botânica; e, o “Hello Navi!”, um aplicativo desenvolvido por alunos da escola Los Frenos nos EUA para auxiliar pessoas com deficiência visual a se deslocarem na escola.

A partir de trabalhos que analisaram a utilização do App Inventor para construção de aplicações educativas (TURBAK; OKERLUND, 2013; WOLBER, 2011), percebe-se que ele implementa requisitos importantes para ferramentas de autoria: permitir a simulação das características dos aplicativos gerados; apresentar uma interface amigável e intuitiva; e, possibilitar a distribuição das aplicações geradas de forma fácil.

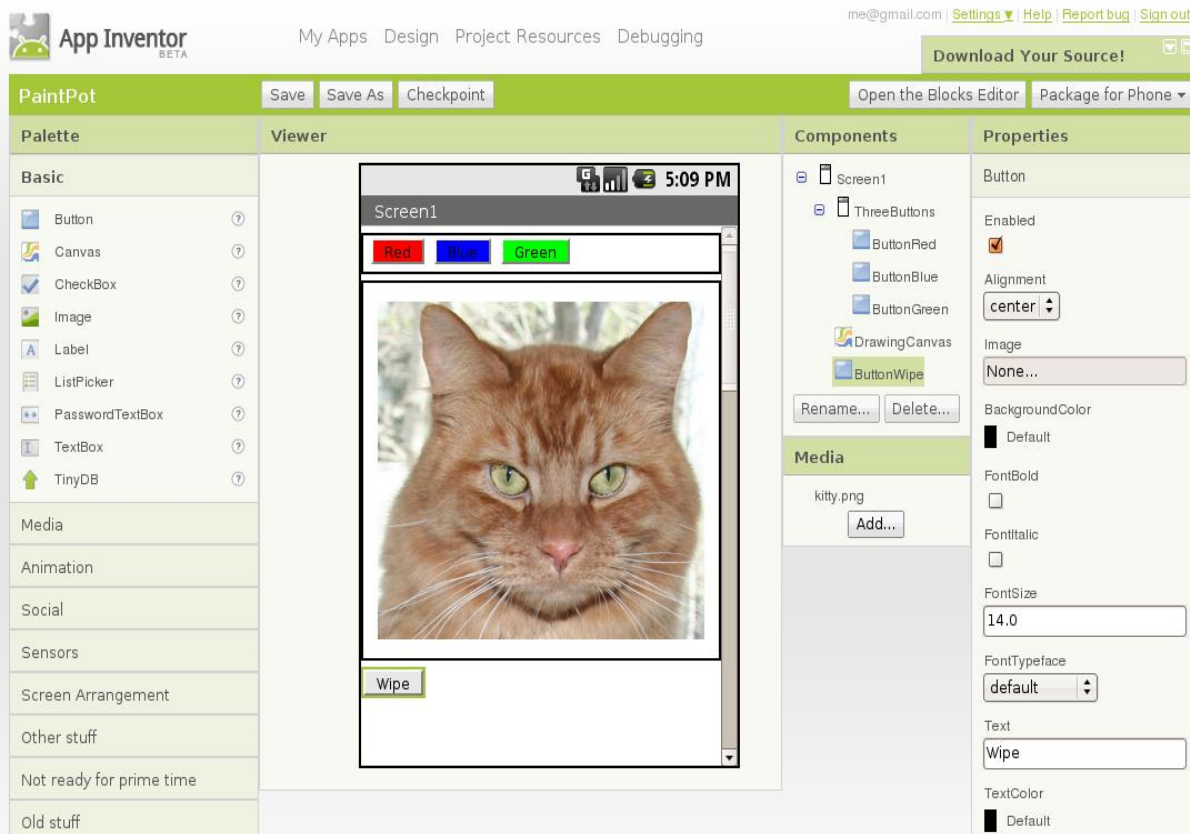
Por outro lado, mesmo o App Inventor apresentando uma interface amigável, o usuário precisa ter conhecimento de alguns conceitos básicos de programação, como manipulação de variáveis e estruturas de condição e repetição. Isso pode se tornar um complicador para utilização por professores que não tenham esses conhecimentos técnicos.

Além disso, o App Inventor não utiliza um modelo padrão aberto para especificação dos processos de ensino e aprendizagem. Isso pode se justificar pelo fato da ferramenta não ter foco exclusivo em aplicativos educativos. Essa restrição dificulta a interoperabilidade com outros softwares educacionais.

Por fim, as aplicações geradas pelo App Inventor têm portabilidade limitada a dispositivos que utilizam a plataforma Android. Portanto, no caso de um professor construir

uma aplicação para utilizar com seus alunos, apenas aqueles que têm dispositivos móveis com esse sistema operacional poderão utilizá-la.

Figura 23 – Tela da ferramenta App Inventor



Fonte: Capturada pelo próprio autor.

O mercado de software para construção de aplicações para dispositivos móveis, educativas ou não, apresenta vários sistemas pagos, tais como: AppMaker, iBuildApp e BuildFire. Entretanto, todas as ferramentas e abordagens apresentadas nas seções anteriores e o foco desse trabalho está em soluções gratuitas e de livre acesso para professores e pesquisadores.

3.6 Análise comparativa

Nesta seção é realizada uma análise comparativa entre as abordagens apresentadas para modelagem, geração e execução de sistemas ubíquos voltados para aulas de campo (Tabela 3). Para escolha dos critérios de comparação foram levados em consideração requisitos presentes em trabalhos com experiências sobre o uso das tecnologias ubíquas em

campo (que serão detalhados no capítulo 4) e características importantes para ferramentas de autoria de aplicações educativas ubíquas apresentadas por Tabuenca *et al.* (2014). São eles:

1. **Modelagem gráfica.** Critério de comparação que indica se a ferramenta permite a criação e edição das aplicações móveis educativas através de modelos gráficos. Inclusive, os projetos criados devem poder ser reutilizados para serem adaptados a situações diferentes das projetadas inicialmente.
2. **Conhecimentos de programação.** Critério que mostra se para utilizar a ferramenta de autoria o usuário necessita ter conhecimentos sobre conceitos de linguagens de programação.
3. **Utilização de Especificações.** Indica se a abordagem proposta utiliza alguma especificação reconhecida de representação de recursos e processos de aprendizagem, como o padrão IEEE *Learning Object Metadata* ou IMS *Learning Design*.
4. **Modelo aberto.** Critério de comparação usado para informar se a abordagem proposta adota um modelo de projeto aberto que permite a interoperabilidade com outros sistemas.
5. **Compartilhamento.** Capacidade da ferramenta de autoria em permitir que os projetos de aplicações criados possam ser compartilhados via Internet para serem reaproveitados por diferentes autores. Para possibilitar esse compartilhamento, é importante a existência de um eficiente mecanismo de busca nos projetos existentes.
6. **Sistema operacional móvel.** Aponta com qual ou com quais sistemas operacionais de dispositivos móveis os aplicativos gerados são compatíveis.
7. **Uso de sensores.** Característica que mostra se a ferramenta de autoria permite que as aplicações projetadas possam utilizar os sensores disponíveis nos dispositivos móveis e quais são eles.
8. **Informação contextual.** Indica se as aplicações móveis têm a capacidade de utilizar informações contextuais obtidas através dos sensores para realizar interações com o usuário.
9. **Dependência de conexão.** Característica que informa se os aplicativos gerados funcionam apenas se os dispositivos móveis estiverem conectados à Internet.
10. **Interação sem fio.** Indica se a ferramenta de autoria prevê recursos para interação entre os participantes através de tecnologias sem fio, como chat via WiFi ou troca de arquivos via Bluetooth.

Tabela 3 – Comparação entre os trabalhos relacionados

Ferramentas	LEMONADE	UoLmP	MAT for ARLearn	Midgar	App Inventor
Critério					
Modelagem gráfica	Sim	Não	Sim	Sim	Sim
Conhecimentos de programação	Não	Sim	Não	Não	Sim
Utilização de Especificações	Não	IMS LD	Não	Não	Não
Modelo aberto	Não	Sim	Não	Não	Não
Compartilhamento	Não	Não	Sim	Sim	Sim
Sistema operacional móvel	Windows	Android	Android	Android	Android
Uso de sensores	GPS	*	GPS	**	GPS, NFC, Bússola e Acelerômetro
Informação Contextual	Localização	*	Localização	**	Localização e Orientação
Dependência de conexão	Não	Sim	Sim	Sim	Não
Interação sem fio	Não	Não	Não	Não	Sim

Fonte: elaborada pelo próprio autor.

* Como a abordagem proposta no trabalho UoLmP (GÓMEZ, 2013) não prevê uma ferramenta de edição, não é possível a inclusão direta no projeto de sensores. Entretanto, a solução teórica apresentada prevê um modelo contextual baseado nas informações de localização, perfil do usuário, tempo, dispositivo móvel e informações ambientais (e.g. temperatura e umidade).

** Os tipos de sensores e as informações contextuais vão depender dos *Smart Objects* disponibilizados no servidor do Midgar (GARCÍA *et al.*, 2014).

A partir da análise dos trabalhos relacionados e considerando os critérios de comparação utilizados na Tabela 3, é possível extrair observações relevantes sobre as soluções para modelagem e geração de aplicações educativas ubíquas. São elas:

- i. Quase todas as abordagens relacionadas, exceto uma, apresentam ferramentas gráficas para modelagem e geração das aplicações. A exceção é o UoLmP (GÓMEZ, 2013). Por utilizarem uma linguagem de modelagem reconhecida (a IMS LD), os autores sugerem que a edição dos projetos seja realizada através de qualquer ferramenta compatível com o modelo, seja com representação gráfica ou textual. Entretanto, além

de colocar uma tarefa a mais para o usuário (escolha do software de edição adequado), a ferramenta escolhida pode não ser a mais apropriada à abordagem proposta.

ii. Duas das abordagens analisadas demandam conhecimentos sobre conceitos de programação: para utilizar a UoLmP (GÓMEZ, 2013) o usuário deverá ter noções de XML e de alguns elementos da linguagem IMS LD; já para utilizar o App Inventor (MIT, 2012) é necessário compreender alguns conceitos relacionados a linguagens de programação, como variáveis e estruturas de repetição e condição. A necessidade desses conhecimentos pode acarretar dificuldade na adoção dessas soluções, implicando na escolha de outras que não demandem tais expertises.

iii. Apenas uma das abordagens apresentadas utilizou uma especificação reconhecida para representação dos recursos e processos de aprendizagem: UoLmP (GÓMEZ, 2013). Com isso, os autores obtiveram os seguintes ganhos: padronização da estrutura dos projetos; possibilidade de reutilização dos modelos; interoperabilidade com ferramentas de edição compatíveis com IMS LD; além da qualidade obtida ao se seguir um modelo estabelecido e aceito por dezenas de instituições em todo o mundo.

iv. Nenhuma das soluções apresentadas gera aplicações móveis para mais de um sistema operacional: quatro geravam apenas para Android e uma apenas para Windows. Tendo em vista a heterogeneidade de dispositivos móveis existentes, a adoção dessas ferramentas restringe o público alvo dos usuários das aplicações geradas.

v. Uma característica comum a todas as abordagens apresentadas é capacidade para gerar aplicações que utilizem os sensores para captura das informações contextuais, com destaque para a localização do usuário. Esta está presente em todos os estudos e pode ser obtida através de diferentes sensores (e.g. GPS ou NFC).

vi. Em três dos trabalhos apresentados, as aplicações nos dispositivos móveis necessitam de conexão com a Internet para funcionar: UoLmP (GÓMEZ, 2013), MAT for ARLearn (TABUENCA *et al.*, 2014) e Midgar (GARCÍA *et al.*, 2014). Essa dependência pode inviabilizar o uso dessas abordagens em determinados tipos de aulas de campo onde não há garantia de conexão com a Internet, por exemplo: explorações a lugares distantes dos centros urbanos ou visitas a cavernas.

vii. Apenas uma das soluções relacionadas possibilita a geração de aplicações móveis com recursos para interação via rede sem fio entre os usuários. Este é um recurso

importante em aulas de campo, porque em alguns momentos os alunos estão dispersos geograficamente e precisam esclarecer dúvidas com o professor ou trocar informações com os colegas.

3.7 Conclusão

Neste capítulo foram descritos trabalhos importantes relacionados à modelagem, geração e execução de sistemas ubíquos voltados para aulas de campo ou que podem ser aplicados nesse tipo de prática educativa. Após as descrições, uma análise comparativa foi realizada para possibilitar uma verificação sobre as características comuns e as diferenças existentes nesse tipo de solução. A partir das constatações citadas e considerando os critérios utilizados, percebe-se que as soluções apresentadas possuem características diferentes e apresentam lacunas importantes para aulas de campo ubíquas. O próximo capítulo apresenta o detalhamento de um estudo sobre os requisitos essenciais para sistemas ubíquos voltados a aulas de campo.

4 REQUISITOS DE SISTEMAS UBÍQUOS PARA AULAS DE CAMPO

Este capítulo apresenta os requisitos que foram usados como base para a construção do ambiente UFC-Inventor. A Seção 4.1 apresenta um mapeamento sistemático realizado a partir de trabalhos publicados entre os anos de 2010 e 2014 com experiências de aulas de campo ubíquas. A Seção 4.2, com base nas práticas observadas no mapeamento, apresenta um conjunto de requisitos importantes para sistemas ubíquos voltados a aulas de campo. Na Seção 4.3, são apresentadas duas aplicações que demonstram o funcionamento dos requisitos identificados de forma integrada. Por fim, a seção 4.4 apresenta as conclusões deste capítulo.

4.1 Mapeamento sistemático

Para a realização deste mapeamento sistemático, foi adotado um processo de revisão baseado em Petersen *et al.* (2008). Esse protocolo prevê cinco passos essenciais a serem seguidos: (i) definição de questões de pesquisa, (ii) realização da pesquisa de estudos primários relevantes, (iii) triagem dos documentos, (iv) busca da *string* nos resumos e textos completos, e (v) a extração das informações e mapeamento. O processo fundamenta-se nas questões de pesquisas e em triagens dos artigos, a partir da *string* de busca e a utilização de critérios sobre o título, o resumo e o artigo em si. As questões de pesquisa que fundamentam esse mapeamento sistemático são:

Q1 – Quais as áreas de conhecimento que estão utilizando as tecnologias ubíquas em campo?

Q2 – Quais os níveis de escolaridade dos alunos que participam das aulas de campo ubíquas?

Q3 – Quais as atividades de aprendizagem estão sendo realizadas através desses sistemas ubíquos?

Q4 – Quais informações contextuais estão sendo usadas nesses sistemas?

Q5 – Em quais Sistemas Operacionais estão funcionando as aulas de campo ubíquas?

Q6 – Quais as formas de transmissão de dados estão sendo utilizadas por essas aplicações?

Foram realizados testes com diferentes combinações de palavras para definir como seria a busca. Optou-se por utilizar uma *string* de busca mais abrangente e aplicar

posteriormente critérios de exclusão, para recuperar um número maior de artigos e minimizar as possibilidades de eliminar algum trabalho relevante. A *string* de busca utilizada foi “*ubiquitous learning*” OR “*pervasive learning*”. Tendo em vista o surgimento frequente de novas tecnologias ubíquas, foram considerados apenas os trabalhos dos últimos cinco anos (entre 2010 e 2014).

Os critérios de exclusão adotados nesse mapeamento sistemático foram:

CE1 - Artigos que não se tratavam de estudos primários;

CE2 - Artigos que não estavam no idioma inglês;

CE3 - Artigos que não apresentavam uma experimentação da pesquisa;

CE4 - Artigos que não apresentavam um ambiente de computação ubíqua voltado para educação com uso fora da sala de aula.

Com esses critérios, objetivou-se manter apenas os artigos que apresentavam sistemas ubíquos voltados para o ensino e a aprendizagem com experimentação utilizando usuários reais e em ambientes fora de sala de aula. A Tabela 4 apresenta as quantidades de artigos que foram identificados, inicialmente agrupados por biblioteca digital. Também exhibe os artigos que foram excluídos ao longo do processo do mapeamento sistemático.

Tabela 4 – Quantidade de artigos resultantes do mapeamento sistemático

Biblioteca Digital	Após a busca inicial	Após 1º Filtro	Após 2º Filtro
ACM Digital Library	151	32	11
IEEE Xplore	187	60	24
Science Direct	263	78	28
Totais	601	170	63

Fonte: elaborada pelo próprio autor.

4.1.1 Análise dos resultados

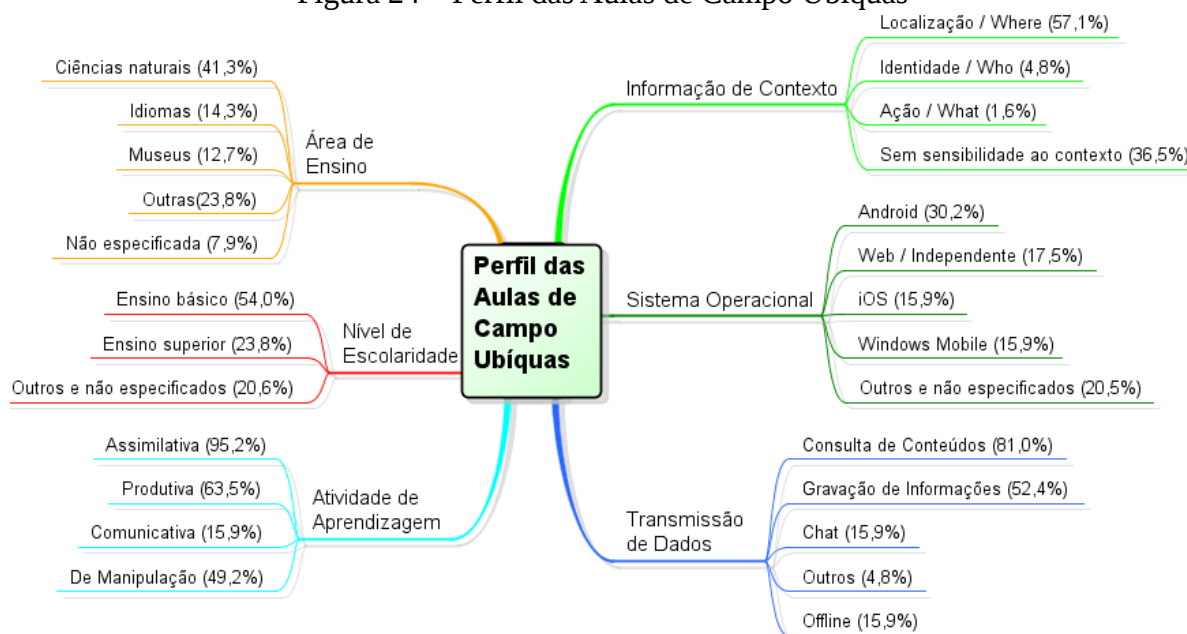
Os sistemas ubíquos aplicados a aulas de campo podem apresentar variedade em termos de: área de ensino, como ciências naturais ou aprendizagem de idiomas; nível de escolaridade, como ensino básico ou superior; mecanismo de sensoriamento, como GPS ou bússola; tecnologia de transmissão de dados sem fio, como consulta online ou envio de arquivos via email; sistema operacional do dispositivo móvel, como Android ou iOS; entre outros. Dessa forma, os profissionais que vão construir ou utilizar esses softwares precisam de informações que fundamentem a escolha das opções mais adequadas às suas realidades.

A partir da análise dos 63 artigos⁵ finais selecionados e considerando as questões de pesquisa propostas, esta seção apresenta os principais resultados desse mapeamento sistemático. A Figura 24 apresenta um mapa mental que mostra uma visão geral do perfil dos sistemas ubíquos utilizados em aulas de campo. Os dados estão agrupados de acordo com as questões de pesquisa estabelecidas nesse mapeamento.

Sobre a primeira questão de pesquisa (Q1), a partir do mapa da Figura 24, observa-se que a área que apresentou a maior quantidade de artigos foi o ensino de ciências naturais (com 41,3%), que envolviam o estudo, por exemplo, de ecologia e biologia. Outras áreas que aparecem em destaque são o ensino de idiomas e as aulas em museus.

Com relação ao nível de escolaridade dos alunos envolvidos (Q2), os dados obtidos indicam que mais da metade dos estudos selecionados (54%) é voltada para jovens do ensino básico. Em 23,8% dos artigos selecionados, as experiências com aulas de campo ubíquas foram realizadas com alunos universitários.

Figura 24 – Perfil das Aulas de Campo Ubíquas



Fonte: elaborada pelo próprio autor.

Para analisar quais eram as principais atividades de aprendizagem utilizadas nas aulas de campo ubíquas (Q3) tomou-se como base o modelo do DialogPlus Toolkit (CONOLE, 2008). Esse modelo apresenta uma taxonomia clara e concisa para auxiliar professores na construção de atividades de aprendizagem suportadas por recursos digitais e estabelece que as

⁵ A lista com os 63 artigos está disponível no APÊNDICE A – Listagem das Referências dos 63 Artigos Analisados no Mapeamento Sistemático desta Tese de Doutorado.

tarefas que os alunos podem executar são agrupadas em seis categorias: assimilativas, produtivas, comunicativas, de manipulação, de simulação e experimentais.

A Tabela 5 mostra as tarefas que compõem as atividades de aprendizagem identificadas nos artigos selecionados (Q3). Elas estão agrupadas de acordo com a categorização proposta no *DialogPlus*. Considerando esse agrupamento, constatou-se um predomínio das atividades assimilativas, que estão presentes em quase todos os trabalhos identificados no mapeamento (95,2%).

Um dos benefícios do uso das tecnologias da computação ubíqua para o ensino é a possibilidade da identificação do contexto em que o aluno se encontra no momento da prática educativa. Para fazer a análise das informações contextuais utilizou-se como base o modelo dos 5 Ws de Abowd e Mynatt (2000): *When* (tempo), *Why* (motivo), *Who* (identidade), *Where* (localização) e *What* (ação). Conforme pode ser observado na Figura 24, a principal informação de contexto utilizada nas aulas de campo ubíquas (Q4) é o local onde o aluno se encontra (*Where*), presente em 57,1% dos estudos. Com percentuais bem menores se encontram as informações contextuais de identidade do usuário (*Who*) e ação realizada (*What*). Além disso, percebeu-se que mais de um terço dos sistemas presentes nos artigos pesquisados (36,5%) não implementam nenhuma forma de sensibilidade ao contexto⁶.

Tabela 5 – Tarefas identificadas no mapeamento agrupadas por categorias

Categoria	Tarefa	Categoria	Tarefa
Assimilativa	Ler textos Ver imagens Assistir a vídeos Escutar áudios Visualizar Mapas	Produtiva	Anotar textos Gravar áudios Gravar vídeos Desenhar na tela Responder questionários
Categoria	Tarefa	Categoria	Tarefa
Comunicativa	Interagir com os alunos Interagir com o professor	De Manipulação	Coletar a localização Capturar fotos Obter a umidade do Ambiente

Fonte: elaborada pelo próprio autor.

O gráfico da

⁶ Consiste na capacidade do sistema em coletar informações contextuais, as quais se referem a todo e qualquer dado que possa ser utilizado para caracterizar uma entidade (pessoa, local ou objeto) relevante para a interação do usuário (DEY; ABOWD, 1999).

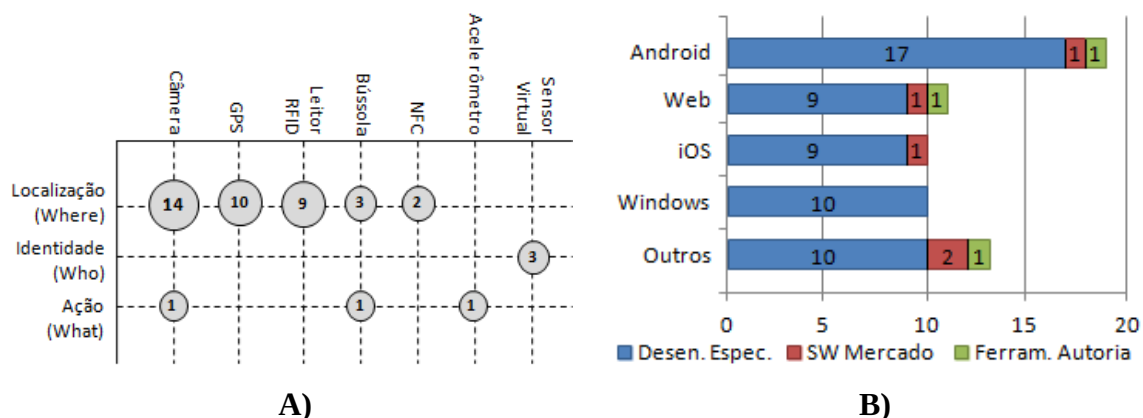
Figura 25-A) mostra a relação entre os sensores adotados e o tipo de informação de contexto utilizada. É importante destacar que na maioria dos estudos, os sensores são utilizados de forma combinada para capturar o contexto do aluno. A partir do gráfico, conclui-se que a câmera digital do dispositivo do aluno foi o sensor mais utilizado, presente em 14 artigos. Em particular, isso ocorreu pelo fato da câmera poder ser usada tanto para captura de imagens quanto para obtenção da localização do aluno através da leitura de QR-codes (códigos de barra *Quick Response*).

Com relação à quinta questão de pesquisa desse mapeamento (Q5), o sistema operacional (SO) Android se destaca como a plataforma de quase um terço dos sistemas ubíquos analisados (30,2%). A segunda plataforma mais utilizada não se trata de um sistema operacional propriamente: 17,5% dos sistemas funcionam em sites Web, que independem de SO. Em seguida estão os sistemas operacionais iOS e Windows Mobile, ambos presentes em 15,9% dos artigos selecionados. Em 20,5% dos trabalhos são utilizados outros sistemas operacionais ou não foram especificados nos artigos.

Além de identificar o Sistema Operacional, buscamos investigar como o sistema ubíquo foi desenvolvido. Como se pode observar no gráfico da

Figura 25-B, a maioria dos sistemas analisados foi desenvolvida especificamente para o experimento mostrado naquele artigo. Apesar de fornecerem poucas informações sobre os aspectos de implementação, foram identificadas como as principais linguagens de programação utilizadas: Java, Objective C, C# e HTML5. Além do desenvolvimento específico dos sistemas, os artigos analisados também utilizavam softwares de mercado já existentes ou ferramentas de autoria para construir os aplicativos a serem utilizados no estudo.

Figura 25 – A) Comparativo entre os sensores e as informações contextuais; B) Comparativo entre os sistemas operacionais e a abordagem de desenvolvimento.



Fonte: elaborados pelo próprio autor.

A última questão analisada nesse mapeamento foi sobre os tipos de transmissão de dados via Internet que eram realizados nas aulas de campo ubíquas (Q6). Nesse aspecto, destacam-se em primeiro e segundo lugar, respectivamente, a consulta (81%) e a gravação (52,4%) de informações. Os dados sobre essa questão indicam conclusões semelhantes com a Q3 (tipos de atividades de aprendizagem). Em ambas as questões destaca-se a consulta a conteúdos, seguida pelo registro das anotações e, com um percentual bem menor, as atividades de interação entre os participantes da aula. Também é importante destacar que, em 15,9% dos estudos, os sistemas não realizavam nenhum tipo de transmissão de dados em campo. Enquadra-se nesse caso, por exemplo, aquela aplicação na qual o conteúdo já está inserido no seu próprio banco de dados.

4.2 Definição dos requisitos

A partir do mapeamento sistemático descrito na seção anterior e das experiências analisadas, esta seção apresenta um conjunto de requisitos identificados como importantes em aulas de campo ubíquas. Com isso, pretendemos estabelecer um guia para suporte ao desenvolvimento de novos sistemas ubíquos para aulas de campo ou para seleção entre softwares existentes. A Tabela 6 apresenta os requisitos funcionais e não-funcionais identificados.

Tabela 6 – Requisitos funcionais e não-funcionais de aulas de campo ubíquas

Requisitos de Aulas de Campo Ubíquas	
Não-Funcionais	Funcionais
• Sensibilidade ao contexto do aluno • Interface multimodal	• Disponibilizar o conteúdo em diferentes mídias • Registrar o material produzido pelo aluno

• Compatibilidade com dispositivos móveis	• Possibilitar interação sem fio entre alunos e professores • Consultar material didático em servidor remoto • Armazenar as informações registradas em servidor centralizado
---	--

Fonte: elaborada pelo próprio autor.

4.2.1 Requisitos não-funcionais

Enquanto executam as atividades em campo, diferenças de conhecimento e comportamento são percebidas entre os estudantes. Nesse momento, ferramentas de computação ubíqua podem ampliar os benefícios da aula de campo ao considerar o contexto no qual o aluno se encontra. A maioria dos estudos analisados (63,5%) nesse mapeamento utiliza alguma informação de contexto durante as atividades de aprendizagem na aula de campo.

Conforme Oviatt (2003), os sistemas com interface multimodal são aqueles que processam duas ou mais formas de entrada do usuário de maneira coordenada, por exemplo, via voz ou toque na tela ou gestos. Esses recursos tornam-se ser importantes durante as aulas de campo, quando os alunos precisam executar diferentes tarefas, como observar o ambiente, escutar as orientações do professor e fazer várias anotações. Além disso, segundo Cohen *et al.* (2015), as interfaces multimodais facilitam a compreensibilidade dos sistemas, evitando treinamentos extensos. Dentre os trabalhos selecionados no mapeamento, diferentes formas de entrada foram utilizadas para capturar a interação do aluno com o sistema, como o microfone e a câmera.

Em todos os estudos analisados no mapeamento sistemático, o aluno utiliza um *smartphone* ou um *tablet* para acessar os recursos do sistema ubíquo. Portanto, tal dispositivo torna-se essencial para viabilizar as aulas de campo ubíquas, independentemente do sistema operacional utilizado. O mapeamento ainda indica que a aplicação pode ser específica para um determinado SO ou independente de plataforma ao funcionar na Web.

4.2.2 Requisitos funcionais

Uma das premissas da aprendizagem multimídia é que os seres humanos possuem canais diferentes para o processamento das informações visuais e auditivas (canal dual) (MAYER, 2002). Assim, um ambiente multimídia para aprendizagem precisa apresentar

conteúdo utilizando diferentes mídias que promovam estímulos visuais e auditivos para que os alunos possam construir representações mentais sobre o que está sendo estudado.

Conforme indicado nesse mapeamento, mais da metade dos artigos selecionados (63,5%) apresentam algum tipo de atividade produtiva. Isto é, os alunos anotam (via texto, áudio, vídeo ou outra forma de entrada) informações importantes para aprendizagem deles, que precisam ser armazenadas pelo sistema. Tendo em vista a variedade e a importância das informações obtidas em campo, é necessário que todos os dados sejam armazenados de forma integrada.

Durante a aula de campo, em diversos momentos os alunos estão dispersos geograficamente e precisam esclarecer dúvidas com o professor ou trocar informações com os colegas. Assim, torna-se um requisito importante para aulas de campo ubíquas a possibilidade dos participantes se comunicarem utilizando as tecnologias sem fio.

Os números apresentados nesse mapeamento demonstram que quase todas as aplicações (95,2%) apresentavam algum tipo de atividade assimilativa. Destacando-se o fato dos conteúdos apresentados combinarem diferentes mídias, predominando-se a associação de textos com imagens. Entretanto, áudios e vídeos também são utilizados. Em algumas dessas atividades de aprendizagem, o material didático necessário não estava disponível no dispositivo móvel do aluno. Nesse momento, torna-se importante que o sistema possa acessar a um servidor remoto para consultar informações complementares. Essa funcionalidade amplia as possibilidades do aluno durante as aulas de campo, dando-lhe mais autonomia durante a prática.

Um dos principais benefícios da aula de campo é possibilitar que as anotações feitas em campo possam ser analisadas posteriormente (ROSLIN *et al.*, 2013). Dessa forma, além de poder registrar as informações de forma integrada, é importante que esses dados sejam persistidos em um servidor centralizado, para consultas posteriores.

Sobre esse requisito, duas estratégias se destacam nos artigos analisados: o registro dos dados online no servidor durante a aula de campo e o armazenamento dos dados localmente no dispositivo móvel para posterior exportação das informações. A escolha da opção mais apropriada passa pela análise da disponibilidade da Internet em campo e da capacidade de armazenamento do dispositivo móvel do aluno.

4.3 Exemplos de implementações dos requisitos

Para fins de demonstração de implementação, são apresentadas duas aplicações e os requisitos identificados na seção anterior: o Geomóvel (MARÇAL *et al.*, 2013) e o CAULS (CHEN; HUANG, 2012). O primeiro trata-se de um software resultado de pesquisas do nosso grupo iniciadas em 2011 (VIANA *et al.*, 2011) com tecnologias ubíquas no ensino de geologia. O segundo foi escolhido considerando público e área de ensino diferentes do Geomóvel, para ampliar o alcance da demonstração proposta nessa seção. Tomando-se como base as características de uma aula de campo ubíqua da Figura 24, a Tabela 7 e a Tabela 8 apresentam respectivamente os perfis do Geomóvel e do CAULS.

O Geomóvel trata-se de um aplicativo Android voltado para auxiliar os alunos do curso de graduação de Geologia em aulas de campo. Dos oito requisitos apontados na seção anterior, apenas a consulta a dados em um servidor não foi implementada no Geomóvel, devido à impossibilidade de conexão com a Internet na maioria das aulas de campo de geologia. Nesse caso, foi usada a base de dados do próprio aplicativo para armazenar o conteúdo necessário.

Tabela 7 – Perfil da aplicação Geomóvel

Característica	Característica
Descrição	Descrição
Área de Ensino: Geologia	Transmissão de dados através de: Envio das anotações por email; e Troca de imagens via Bluetooth.
Nível de Escolaridade: Ensino Superior	
Informações de Contexto: Localização (via GPS)	
Sistema Operacional: Android	
Atividades de Aprendizagem:	
Produtiva. Ex: Registro dos dados do afloramento (ponto de estudo);	
Assimilativa. Ex: Consulta a instruções sobre medições geológicas;	
Comunicativa. Ex: Troca de imagens capturadas entre os alunos;	
De Manipulação. Ex: obtenção das coordenadas geológicas.	

Fonte: elaborada pelo próprio autor.

Chen e Huang (2012) apresentam o CAULS (context-aware *ubiquitous learning system*), um sistema para alunos do ensino fundamental utilizarem durante visitas a museus,

com um estudo de caso em história aborígine de Taiwan. Dentre os identificados na seção anterior, o único requisito não implementado pelo CAULS é a possibilidade de interação entre os participantes da aula no museu através do sistema. O sistema CAULS implementa todos os requisitos relacionados à transmissão de dados tendo em vista estar baseado na utilização da conexão sem fio do museu.

Tabela 8 – Perfil da aplicação CAULS

Característica	Característica
Descrição	Descrição
Área de Ensino: História	Transmissão de dados através de: Registro da localização no servidor; Gravação de avaliações no servidor; Consulta de dados no servidor.
Nível de Escolaridade: Ensino Fundamental	
Informações de Contexto: Localização (via RFID ⁷)	
Sistema Operacional: Windows Mobile	
Atividades de Aprendizagem:	
Produtiva. Ex: Gravação das respostas a questionários de avaliação no servidor;	
Assimilativa. Ex: Consulta de informações sobre os artefatos aborígenes no servidor;	
De Manipulação. Ex: obtenção da localização do aluno através das etiquetas RFID.	

Fonte: elaborada pelo próprio autor.

4.4 Conclusão

Diversos pesquisadores têm conduzido experiências acerca do uso das tecnologias ubíquas aplicadas em aulas de campo. A partir de um mapeamento sistemático sobre essas pesquisas, foi identificado um conjunto de requisitos funcionais e não funcionais importantes para aulas de campo ubíquas. Para a elaboração dessa lista, foi realizado um levantamento com artigos na área publicados entre os anos de 2010 e 2014. Pretendeu-se, com esse capítulo, definir o perfil dos sistemas ubíquos utilizados em aulas de campo.

⁷ RFID é um método de identificação automática por radiofrequência através de leitores e etiquetas no padrão RFID ("Radio-Frequency Identification").

A título de demonstração, foram apresentados os perfis de dois diferentes sistemas ubíquos para apoio a aulas de campo. Com isso, foi possível mostrar duas possibilidades distintas de implementação dos requisitos identificados. O próximo capítulo apresenta o detalhamento do ambiente UFC-Inventor e todos os seus componentes.

5. O AMBIENTE UFC-INVENTOR

Este capítulo descreve o ambiente computacional proposto neste trabalho para modelagem e geração de aplicações ubíquas com foco em aulas de campo: o UFC-Inventor. A seção 5.1 apresenta uma visão geral do ambiente, descrevendo suas principais características. Na seção 5.2 é apresentada a linguagem específica de domínio ML4UL, que é o núcleo do ambiente. As seções 5.3 e 5.4 tratam, respectivamente, dos dois módulos do ambiente: UFC-GLM e UFC-Generator. Por fim, a seção 5.5 conclui este capítulo.

5.1 Visão geral

Conforme apresentado na seção 1.1, as aulas de campo são importantes práticas educativas que têm seus benefícios ampliados quando aliadas às tecnologias ubíquas, através de recursos como: adaptação contextualizada das atividades e conteúdos em campo, interação entre alunos e professores através de redes sem fio e consulta a material didático em campo. Como mencionado no Capítulo 4, utilizou-se o termo aula de campo ubíqua (ou a sigla UFC – *Ubiquitous Field Class*) para referenciar a prática educativa em campo suportada pelas tecnologias ubíquas. Diversos trabalhos apontam as vantagens obtidas com as UFCs (CHIANG; YANG; HWANG, 2014; MARÇAL *et al.*, 2014; WU; HWANG; TSAI, 2013).

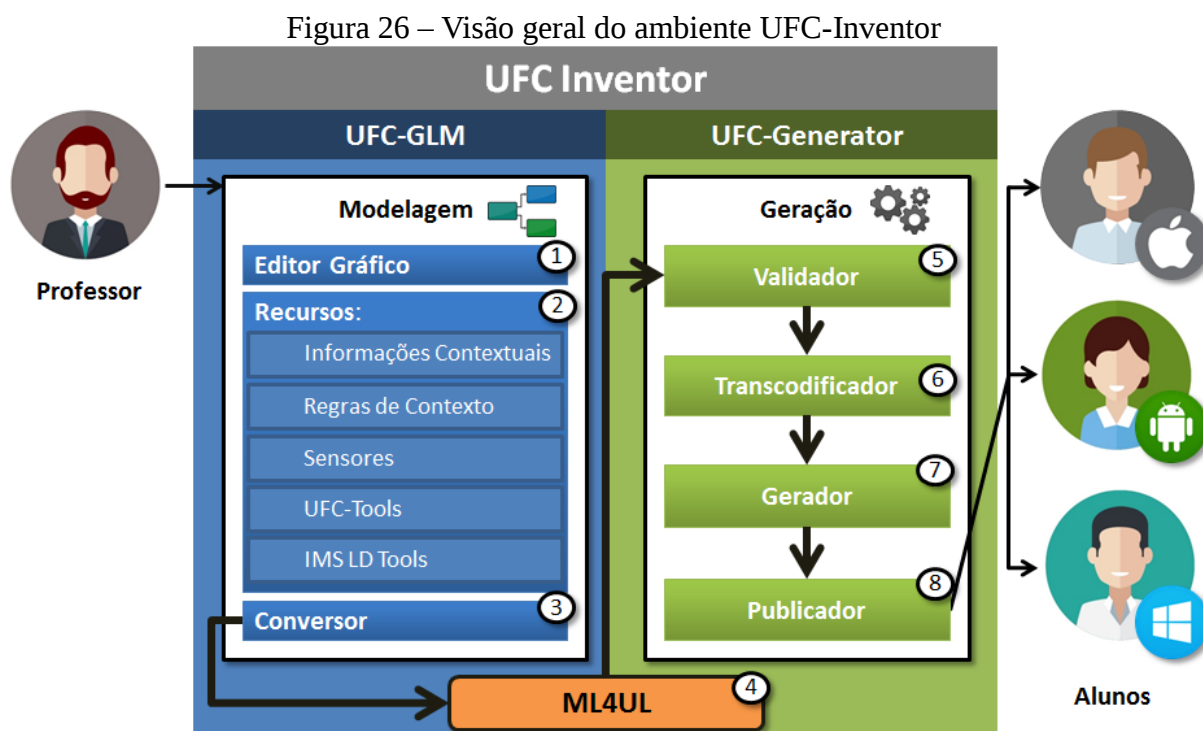
Considerando as três fases de uma aula de campo (pré-campo, campo e pós-campo) e a adição de tecnologias ubíquas, propõe-se o UFC-Inventor, um ambiente para representação e geração de sistemas para UFCs. Através dele é possível modelar as aulas de campo ubíquas (pré-campo) e gerar aplicações ubíquas para serem usadas durante as aulas em campo, as quais permitirão a coleta e armazenamento de informações para análises posteriores (pós-campo).

A

Figura 26 apresenta uma visão geral do ambiente UFC-Inventor, o qual é dividido em dois módulos principais: o UFC-GLM, sistema responsável pela modelagem gráfica das UFCs e o UFC-Generator, responsável pela geração das aplicações ubíquas. A interação entre os dois módulos ocorre através da especificação em formato ML4UL da aula de campo.

O módulo UFC-GLM consiste em uma ferramenta de autoria onde o projetista (e.g. professor ou especialista de área ou outro profissional de ensino) pode realizar as seguintes ações: modelar, através do editor gráfico (1), as atividades de aprendizagem e o fluxo que elas devem acontecer; inserir na especificação recursos de computação ubíqua, como sensores e

comunicação sem fio (2); e, converter (3) o modelo gráfico elaborado com os recursos adicionados para uma especificação no formato ML4UL (4).



Fonte: elaborada pelo próprio autor.

O arquivo com a codificação da aula de campo ubíqua em ML4UL é a entrada para a ferramenta UFC-Generator poder gerar as aplicações ubíquas a serem usadas em campo. Para isso, a especificação deve passar por um módulo validador (5) que faz a leitura do arquivo de entrada e verifica a sua conformidade. Posteriormente, o módulo Transcodificador (6) transforma a especificação da aula de campo de ML4UL para código-fonte em uma linguagem específica. Foi adotada a plataforma Apache CordovaTM ⁸. Depois, o UFC-Generator executa o Gerador (7), que nessa implementação trata-se do compilador CordovaTM. Por fim, o módulo Publicador (8) identifica o aplicativo gerado e o disponibiliza para os alunos poderem instalá-lo em seus dispositivos móveis.

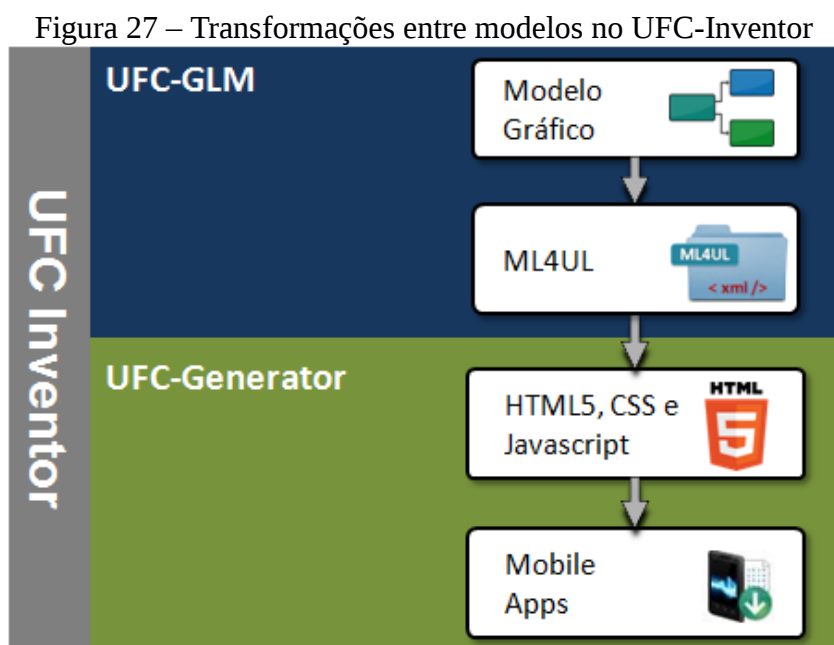
A escolha da plataforma CordovaTM aconteceu em razão dos seguintes fatores: ser um framework desenvolvido e mantido por uma instituição com vasta experiência no desenvolvimento de sistemas, *Apache Software Foundation*⁹; ser uma plataforma livre e de código aberto; ter uma grande comunidade de desenvolvedores e *plug-ins* com suporte a inúmeros recursos; utilizar linguagens de programação e padrões amplamente adotados (HTML, Javascript e CSS); e ser compatível com os principais sistemas operacionais para

⁸ <https://cordova.apache.org/>

⁹ <http://apache.org/>

dispositivos móveis (Android, iOS, Windows Phone, e Blackberry). Atualmente, a Cordova™ conta com mais de 860 *plug-ins* que permitem que sejam desenvolvidas aplicações que suportem os mais variados recursos dos dispositivos móveis, tais como: câmera, acelerômetro, bússola, transmissão de arquivos, dados do dispositivo (e.g. bateria, contato, agenda), entre outros. Além disso, a plataforma possibilita que novos plug-ins sejam desenvolvidos, a medida que novas tecnologias sejam incorporadas aos dispositivos móveis. Uma lista completa com os plug-ins disponíveis atualmente pode ser acessada no site da plataforma Cordova™.

A solução proposta através do UFC-Inventor segue a abordagem do desenvolvimento dirigido por modelos (BRAMBILLA; CABOT; WIMMER, 2012), descrita no Capítulo 2. A Figura 27 mostra as transformações entre os modelos que acontecem no ambiente UFC-Inventor. Parte-se de uma especificação de mais alto nível e independente de plataforma (diagrama da aula de campo ubíquo), então, esse modelo gráfico é convertido para uma codificação textual em ML4UL. Posteriormente, essa especificação em ML4UL é transformada em um modelo baseado em código-fonte de linguagem de programação (HTML5 e Javascript). Finalmente, a última transformação é realizada e obtém-se o arquivo com a codificação binária da aplicação móvel.



Fonte: elaborada pelo próprio autor.

A utilização da abordagem de desenvolvimento dirigido por modelos traz benefícios importantes para o ambiente UFC-Inventor. São eles:

- O modelo gráfico possibilita uma representação em alto nível da aula de campo ubíqua através de elementos que estão mais próximos dos conceitos do domínio (e.g. atividades e fluxo de aprendizagem). Dessa forma, exige-se do projetista pouco ou nenhum conhecimento sobre linguagem de programação para gerar as aplicações ubíquas.
- Ao manter a estrutura do IMS LD, a ML4UL herda todas as características dessa especificação, além de contar com os novos elementos focados nas tecnologias ubíquas. Assim, a representação da aula de campo ubíqua seguirá um modelo reconhecido no meio acadêmico.
- O uso da linguagem específica de domínio ML4UL, como um modelo intermediário entre o diagrama e o código-fonte, proporciona uma diminuição da dependência da tecnologia. Com isso, por exemplo, no caso da descontinuidade da linguagem de programação utilizada (HTML5 e Javascript), a mudança será necessária apenas na função de transcodificação do módulo UFC-Generator. Nessa situação, o Transcodificador deverá transformar a especificação em ML4UL para uma codificação em outra linguagem e em seguida chamar o respectivo compilador.
- A estrutura do UFC-Inventor, baseada em dois softwares distintos, facilita a interoperabilidade com outras ferramentas. Por exemplo, um editor gráfico de terceiros pode ser utilizado para modelar a aula de campo ubíqua, gerar a especificação em ML4UL e chamar o módulo UFC-Generator para gerar as aplicações ubíquas. De forma oposta, pode-se usar o módulo UFC-GLM para especificar a aula de campo e, depois, a codificação em ML4UL servir com entrada para outra ferramenta de geração de aplicações ou um software interpretador no próprio dispositivo móvel.
- O fato de o UFC-Inventor gerar também o código-fonte e não apenas o arquivo binário do aplicativo móvel é outra característica relevante do ambiente. Com isso, através da reutilização e alteração da codificação gerada, outras aplicações com requisitos diferentes dos projetados podem ser desenvolvidas. É importante destacar dois aspectos sobre essa característica: ela é uma funcionalidade do UFC-Inventor voltada para programadores que detém conhecimentos sobre as linguagens nas quais as aplicações são geradas (HTML 5, Javascript e CSS); e, esse código gerado pode ser alterado através de uma ferramenta de programação externa ao UFC-Inventor, porém essas alterações não serão refletidas no modelo original.

5.2 A linguagem ML4UL

Essa seção apresenta a ML4UL (*Modeling Language for Ubiquitous learning*), uma linguagem específica de domínio voltada para representação das aulas de campo ubíquas através de uma especificação aberta, reutilizável e legível por computador. A ML4UL é uma DSL para especificação de sistemas ubíquos aplicados a aulas de campo, a qual foi estendida a partir da linguagem IMS LD. Dessa forma, todas as propriedades da DSL base, a IMS LD, permanecem disponíveis para a nova linguagem, a ML4UL (VAN DEURSEN; KLINT; VISSER, 2000).

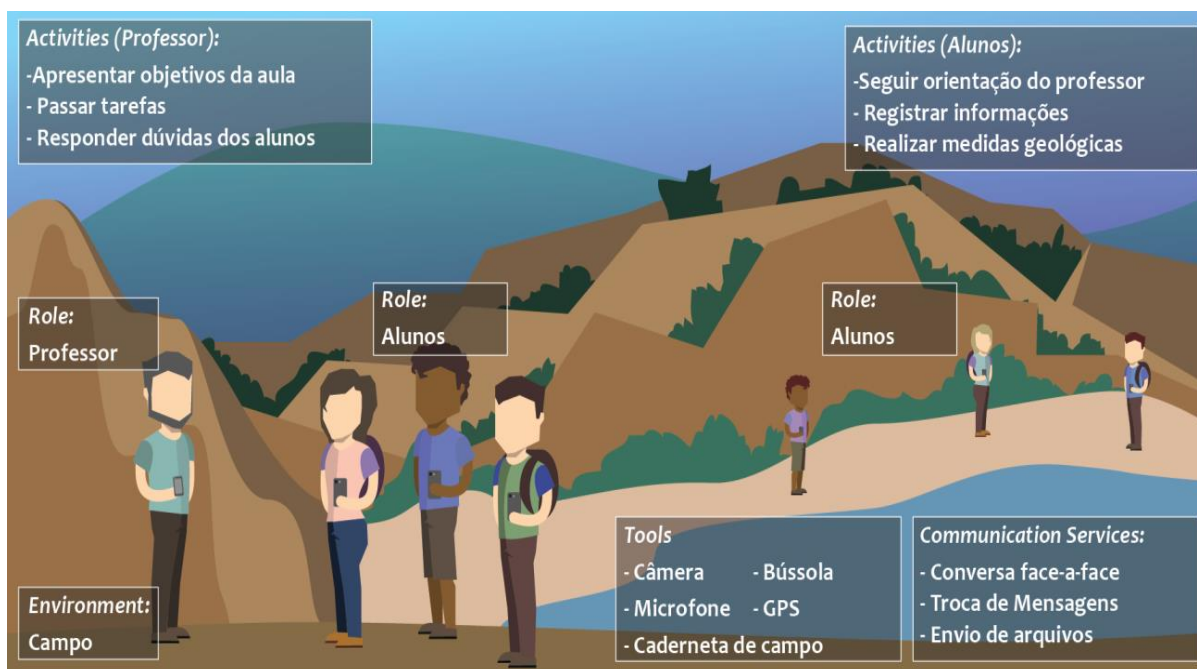
5.2.1 Cenário

Durante a execução de uma aula de campo, os alunos experimentam no ambiente real o que aprenderam em sala de aula e são estimulados a realizar inúmeras atividades, que irão variar de acordo com a área estudada e o público alvo. No contexto desse trabalho, com o propósito de ilustrar um cenário para a abordagem proposta será utilizada uma aula de campo sobre Geologia. Para auxiliar a visualização desse cenário, foi elaborada a Figura 28, que ilustra uma aula de campo com professor e alunos utilizando *smartphones*. Para composição dessa imagem, tomamos como base as características do sistema ubíquo apresentado em (MARÇAL *et al.*, 2014).

As caixas de textos da Figura 28 exibem um mapeamento entre os elementos do ambiente real e aqueles presentes na linguagem IMS LD. O cenário ilustrado apresenta elementos que não precisam das tecnologias ubíquas, como a atividade “Apresentar objetivos da aula”, e outros que precisam, como o sensor “GPS”.

Nesse cenário, existem dois elementos IMS LD do tipo *<role>* (professor e aluno), que representam os participantes da unidade de aprendizagem. Cada um deles pode executar *<activities>*, que equivalem às atividades de aprendizagem que os alunos podem executar e aquelas de suporte realizadas pelo professor. O local onde ocorre a aula de campo é codificado em IMS LD através do elemento *<environment>*, que nesse exemplo é um campo. Entre os elementos que podem compor um *<environment>*, têm-se: as ferramentas (*<tool-objects>*) disponíveis; e os serviços para a comunicação entre os participantes (*<communication-services>*).

Figura 28 – Ilustração de uma aula de campo com *smartphones* e elementos IMS LD



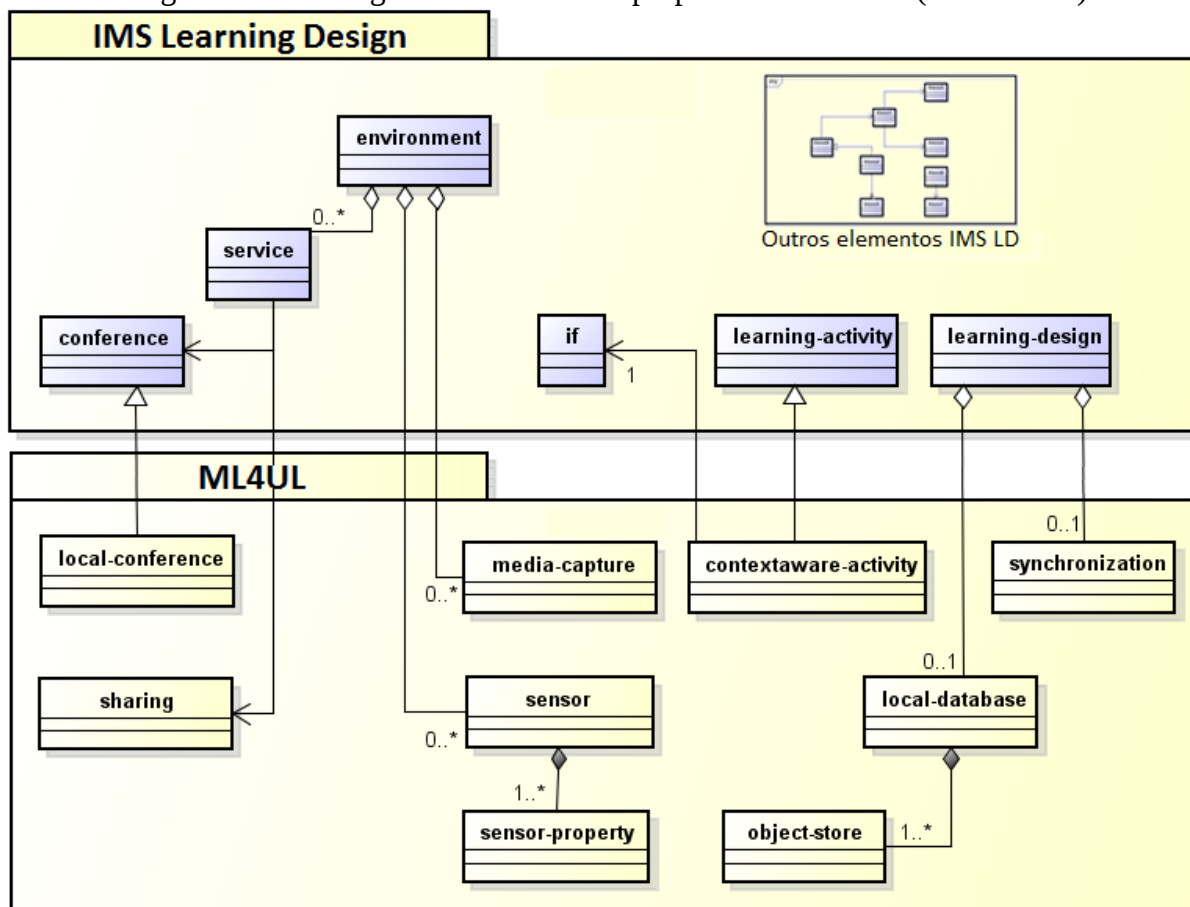
Fonte: elaborada pelo próprio autor.

5.2.2 Especificação da ML4UL

A ML4UL pode ser vista como um metamodelo que tem como premissa fundamental a manutenção da estrutura da especificação IMS LD. A Figura 29 apresenta os principais elementos da IMS LD, dentro do retângulo superior. Abaixo, estão os novos elementos propostos na ML4UL: local-conference, sharing, media-capture, sensor, sensor-property, contextaware-activity, local-database, object-store e synchronization. As alterações e adaptações propostas pela ML4UL não afetam o modelo IMS LD. Dessa forma, pretende-se manter todos os benefícios da especificação original e adicionar novos elementos, com foco nas tecnologias ubíquas aplicadas a aulas de campo.

Um aspecto importante mantido pela ML4UL é o modelo de empacotamento da IMS LD, o IMS Content Packing. Assim, o projeto de uma aula em ML4UL é composto por: um arquivo em formato XML (*imsmanifest.xml*), que contém a especificação da unidade de aprendizagem; e, os arquivos que são necessários para execução daquela UoL, como imagens e vídeos. As próximas subseções apresentam os novos elementos previstos na ML4UL.

Figura 29 – Visão geral dos elementos propostos na ML4UL (em amarelo)



Fonte: elaborada pelo próprio autor.

5.2.2.1 Serviços de comunicação

Durante alguns momentos da aula de campo, os alunos estão dispersos geograficamente e precisam esclarecer dúvidas com o professor ou trocar informações com os colegas. As tecnologias ubíquas têm sido utilizadas para ampliar essa comunicação entre alunos e professores em campo, seja via chat textual (CHIANG; YANG; HWANG, 2014) ou via vídeo (STEFAN *et al.*, 2013).

Dessa forma, a ML4UL conta com o elemento *<local-conference>*, que é uma especialização do elemento *<conference>* da IMS LD. Através desse novo elemento será possível especificar: a atividade de chat local, o tipo de tecnologia sem fio a ser utilizada (como *Bluetooth* ou *Wi-Fi Direct*), os parâmetros para essa conexão; e, de que forma será a comunicação entre os participantes (e.g. texto, áudio ou vídeo).

Um recurso para comunicação utilizado para sistemas de *mobile learning* são as mensagens instantâneas dos dispositivos móveis (serviço conhecido como *Mobile Instant Messaging*). Essas mensagens, diferentemente dos *chats*, não pressupõem uma sessão para interação entre todos os participantes. Outra importante ferramenta de interação entre participantes de uma aula de campo é a possibilidade da troca de arquivos (como fotos ou gravações). A IMS LD não dispõe de um elemento que permita a representação nem das mensagens instantâneas nem do compartilhamento de arquivos entre os participantes. Portanto, a ML4UL tem como solução para essas limitações o elemento *<sharing>*. Através dele, é possível especificar o compartilhamento de mensagens ou arquivos entre os envolvidos na aula de campo.

5.2.2.2 Ferramentas de captura e sensores

Um requisito importante presente em pesquisas sobre o uso dos sistemas ubíquos em aulas de campo é a captura de dados através de sensores (MARÇAL *et al.*, 2013; WU; HWANG; TSAI, 2013). A obtenção dessas informações é importante para indicar o contexto em que o aluno se encontra e permitir que, a partir delas, decisões sejam tomadas. Considerando que a IMS LD não fornece suporte a modelagem desse tipo de recurso, a ML4UL conta com dois novos elementos para solucionar essa questão: *<media-capture>* e *<sensor>*. Ambos têm atributos para indicar as configurações dos sensores e para receber os dados capturados.

O *<media-capture>* é o elemento responsável por permitir a especificação dos mecanismos de captura de imagens, áudios e vídeos. Alguns exemplos de atributos desse elemento são *<media-type>* e *<media-file>*, que indicam respectivamente o tipo de mídia a ser obtida (imagem, vídeo ou áudio) e o caminho do arquivo que armazenará a mídia capturada.

Através do elemento *<sensor>* é possível modelar o funcionamento dos sensores relacionados aos movimentos realizados pelos alunos em seus dispositivos móveis, a localização em que eles se encontram e às variáveis ambientais naquele instante, por exemplo: bússola, GPS e luminosidade. O elemento *<sensor>* é composto de elementos *<sensor-property>*, que são usados para especificar as propriedades do sensor, como latitude e longitude no caso do GPS.

5.2.2.3 *Persistência local dos dados*

A IMS LD assume que as interações dos participantes entre si e com o ambiente virtual de aprendizagem ocorrem através da Internet. Entretanto, no caso das aulas de campo, em diversos momentos os alunos não têm conexão e precisariam interagir com o sistema ubíquo, por exemplo, para armazenar dados sobre o local em que se encontram. Através da IMS LD não é possível especificar essa funcionalidade.

Para resolver essa questão da gerência dos dados locais, a solução através da ML4UL é baseada em dois elementos. O *<object-store>*, que é responsável por representar o armazenamento de um conjunto de registros. Cada registro consiste de uma chave e um valor. O outro elemento proposto é o *<local-database>*, que consiste em um ou mais elementos do tipo *<object-store>*. Cada Unidade de Aprendizagem pode ter um único *<local-database>* que pode conter um ou vários *<object-store>*.

Os principais motivos para escolha desta solução foram: essa abordagem é baseada em uma especificação de armazenamento de dados local do W3C (World Wide Web Consortium), a Indexed Database API (IndexedDB) ; essa API é uma especificação para linguagem HTML5 e compatível com vários browsers web e sistemas operacionais de dispositivos móveis; por último, a IndexedDB apresenta uma estrutura mais simples que os modelos de banco de dados relacionais.

5.2.2.4 *Sensibilidade ao contexto*

Além de serviços de comunicação e recursos de sensoriamento, os sistemas ubíquos podem fazer uso das informações contextuais para ampliar os benefícios das aulas de campo. No caso da aula de geologia do cenário apresentado, um exemplo de informação de contexto importante é a localização do aluno. A partir dela, um sistema ubíquo pode, por exemplo, fornecer vídeos informativos e conteúdo didático sobre um determinado afloramento no qual o aluno se encontra ou notificar o aprendiz de uma atividade que precisa ser realizada naquele contexto.

De acordo com Derntl e Hummel (2005), existem três possíveis formas nas quais as informações contextuais afetam o fluxo das atividades de aprendizagem e vice-versa, que acontecem quando:

- i. O valor da informação contextual é alterado pela atividade;
- ii. O contexto modifica o funcionamento da atividade;

- iii. O fluxo das atividades de aprendizagem depende do valor da informação contextual.

Burgos (2012) afirma que representar atividades sensíveis ao contexto utilizando o modelo original IMS LD não é uma tarefa simples e direta. Desta forma, é proposto na ML4UL uma solução, baseada nas propostas de Gómez et al. (2014) e Mavroudi e Hadzilacos (2012), para representação de atividades de aprendizagem sensíveis ao contexto com comportamento adaptativo.

Nos casos em que o contexto é alterado pela atividade ou altera o funcionamento dela, deve-se utilizar os elementos *<glob-property>*, *<change-property-value>* e *<property-ref>*, que permitem o armazenamento, a alteração e a consulta dos valores das informações contextuais, respectivamente. Por exemplo, esses elementos podem ser usados para registrar cada um dos locais visitados pelo estudante em uma aula.

Na situação em que o fluxo das atividades de aprendizagem é alterado de acordo com os valores das informações contextuais, utiliza-se o elemento *<contextaware-activity>*. Este tem como objetivo permitir de forma direta a especificação de atividades adaptativas e sensíveis ao contexto. Trata-se de uma especialização do elemento *<learning-activity>*, com um atributo adicional e obrigatório: o elemento *<if>*. Então, uma atividade do tipo *<contextaware-activity>* só deve ser executada se as condições especificadas no seu elemento *<if>* forem satisfeitas. Por exemplo, em uma visita a um museu, o fluxo de aprendizagem da UoL aponta para uma atividade sensível ao contexto que exibe de um vídeo sobre as múmias do Egito. Entretanto, ela só deve ser executada se o estudante tiver visitado a sala do museu sobre o Egito. Caso contrário, essa atividade (exibir o vídeo) não é executada e o fluxo segue para a próxima atividade.

5.2.2.5 Sincronização dos dados

Um benefício importante de uma aula de campo é a possibilidade dos dados coletados serem utilizados e analisados após o retorno do campo (QUINN, 2015; MEDZINI, MEISHAR-TAL, SNEH, 2015; BURSZTYN et al., 2015). Considerando isso e a existência na ML4UL de elementos para especificação da persistência dos dados registrados localmente, torna-se essencial que a linguagem disponibilize mecanismos para que as informações anotadas em campo possam ser disponibilizadas em outro meio que não apenas o dispositivo móvel do estudante.

Para isso, a ML4UL conta com o elemento *<synchronization>*, que possibilita a codificação da sincronização dos dados registrados na aplicação móvel para um servidor de banco de dados remoto. Desta forma, essa solução assume a existência de um servidor na Internet para onde serão enviados e armazenados os dados coletados em campo. Assim, o *<synchronization>* é composto por atributos de configuração e transmissão, como o endereço do servidor na Internet e as informações de acesso (usuário e senha).

Tendo em vista que a ML4UL permite a especificação de diferentes aulas de campo ubíquas e que os dados coletados irão variar de acordo com as necessidades e objetivos de cada aula, não é possível estabelecer um modelo único de banco de dados no servidor. Assim, utilizou-se uma solução baseada em uma única tabela genérica para sincronização dos dados das diferentes aplicações geradas.

A Tabela 9 mostra as informações dos campos que compõem a tabela ‘Registros’, que armazenará as informações sincronizadas no servidor. Pode-se observar que ela é formada por sete campos, sendo a chave primária composta por cinco. A coluna ‘Descrição’ apresenta as informações sobre cada um dos campos. Para facilitar a compreensão sobre como pode se dar o armazenamento das informações, a

Tabela 10 apresenta registros fictícios para exemplificar. Os registros mostrados nessa tabela constituem informações geradas por um único criador (‘000001’). Dois dispositivos móveis diferentes sincronizaram as informações com o servidor (‘8ae54545’ e ‘9u734jh4ik’), cada um com um registro único (‘1’ e ‘2’). As outras informações estão relacionadas às características específicas da aplicação (‘Nome Aluno’, ‘Pergunta 1’ e ‘Localização’).

Tabela 9 – Tabela 'Registros' para armazenamento das informações em servidor na Internet

Campo	Tipo	Chave	Descrição
id_criador	<i>String</i>	S	Identificador único do criador (e.g. professor, projetista) da aplicação.
sr_dispositivo	<i>String</i>	S	Número de série ou outro identificador do dispositivo móvel que registrou as informações.
dh_registro	data-hora	S	Data e hora em que o registro foi gravado na aplicação móvel.
sq_registro	Sequencial	S	Campo sequencial para cada registro gravado no dispositivo móvel.
nm_campo	<i>String</i>	S	Nome do campo utilizado na aplicação móvel.

vr_campo	<i>String</i>	-	Valor do campo gravado no dispositivo móvel.
tp_campo	<i>String</i>	-	Tipo do campo.

Tabela 10 – Exemplo de preenchimento da tabela 'Registros' com dados fictícios

id_criador	sr_dispositivo	dh_registro	sq_registro	nm_campo	vr_campo	tp_campo
000001	8ae54545	1435301115	1	Nome Aluno	João	<i>string</i>
000001	8ae54545	1435301115	1	Pergunta 1	Resposta abc	<i>string</i>
000001	8ae54545	1435301115	1	Localização	-3.740410, -38.4933	localização
000001	9u734jh4ik	1840011215	2	Nome Aluno	Maria	<i>String</i>
000001	9u734jh4ik	1840011215	2	Pergunta 1	Resposta xyz	<i>String</i>
000001	9u734jh4ik	1840011215	2	Localização	-3.745818, -38.575037	localização

É importante destacar que a tabela ‘Registros’ tem o objetivo de funcionar como um repositório genérico dos dados coletados pelas aplicações móveis baseadas na ML4UL. Assim, a linguagem não especifica como essas informações serão utilizadas, ficando essa tarefa a cargo da equipe que desenvolverá o sistema que acessará os dados anotados em campo.

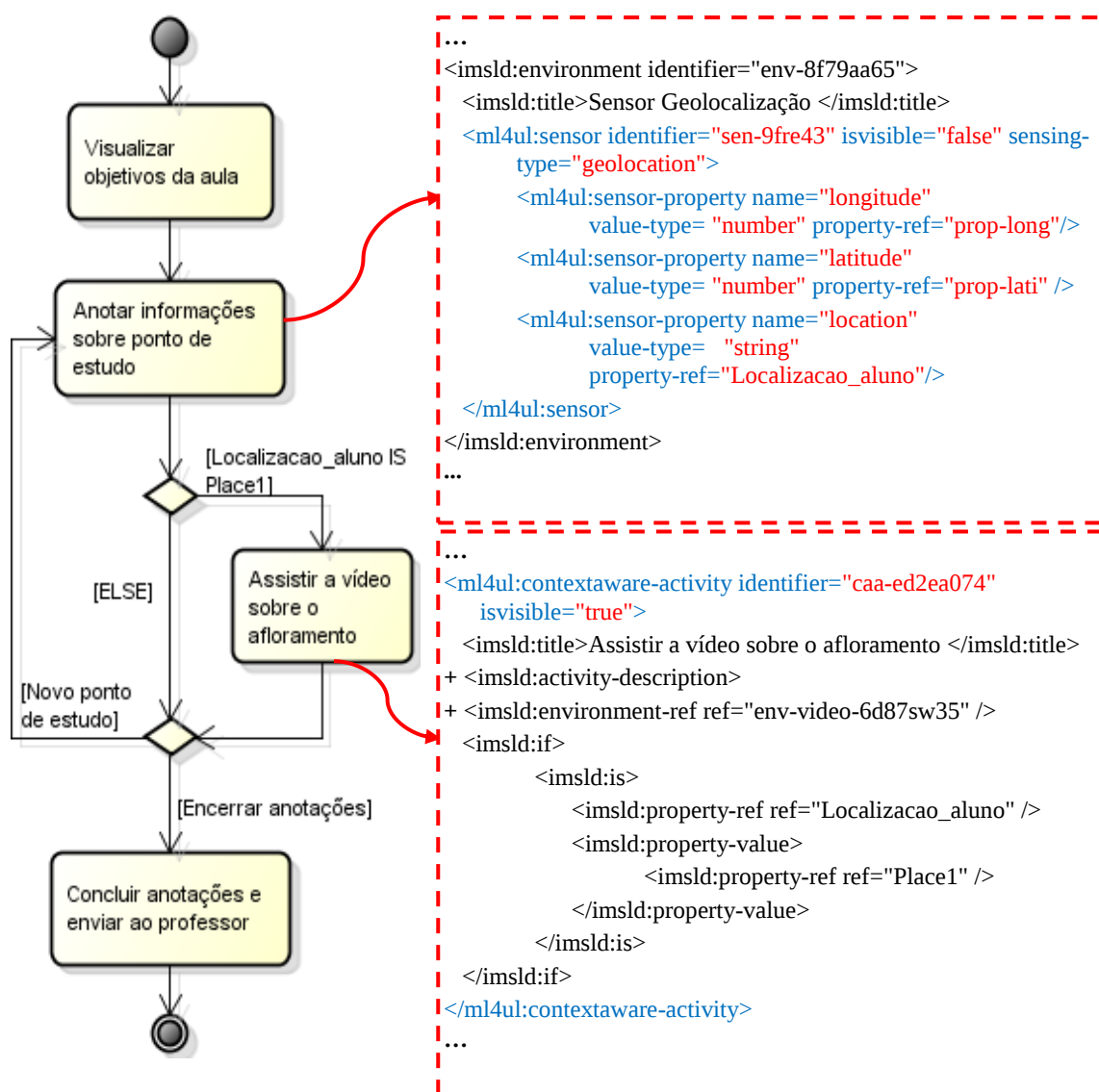
5.2.2.6 Exemplo de uso da ML4UL

Para demonstrar um exemplo de uso da ML4UL, utilizaremos como cenário uma aula de campo de geologia. Nesse exemplo, o aluno deve inicialmente utilizar o dispositivo móvel para visualizar as informações e os objetivos da aula. Posteriormente, ele deve anotar informações (via texto, áudio ou foto) sobre aquele ponto que está estudando. Em seguida, se ele estiver em uma determinada região, ele poderá assistir a um vídeo sobre o afloramento em que ele se encontra localizado. Por fim, o aluno pode anotar mais observações e depois deve transmitir em campo as anotações ao professor.

A Figura 30 mostra um diagrama de atividades da UML que representa graficamente a aula utilizada como exemplo e trechos da especificação em IMS LD e ML4UL. Na codificação, os elementos da IMS LD e da ML4UL têm como prefixos os *namespaces imslld* e *ml4ul*, respectivamente. As linhas de código relativas à ML4UL estão em azul e vermelho. Tendo em vista o fato das especificações em IMS LD serem muito extensas e buscando uma

simplificação da explicação, são apresentados os códigos de apenas dois elementos da ML4UL: *<sensor>* e *<contextaware-activity>*. Entretanto, para representar a aula do exemplo também são necessários os seguintes outros elementos: *<media-capture>* para indicar a captura de foto e áudio; e *<sharing>* para especificar o compartilhamento em campo das informações anotadas.

Figura 30 – Exemplo em Diagrama de Atividades e codificação em IMS LD com ML4UL



Fonte: elaborada pelo próprio autor.

O primeiro trecho de codificação exemplifica a especificação do elemento *<sensor>* da ML4UL. São mostrados: o atributo “*sensing-type*”, que indica o tipo de sensor a ser usado; e o elemento *<sensor-property>*, usado para indicar quais as propriedades deverão receber os valores de retorno após a execução do sensor. Considerando esse código, o programa

interpretador de IMS LD com ML4UL deverá chamar o sensor de geolocalização, obter as coordenadas e atualizar os valores nas propriedades especificadas.

O segundo código extraído mostra o elemento da ML4UL *<contextaware-activity>*. Nele, além dos elementos padrões do *<learning-activity>* da IMS LD, como *<activity-description>* e *<title>*, aparece o elemento *<if>*. Na condição mostrada nesse exemplo, é verificado se a propriedade “*Localizacao_aluno*”, obtida através do elemento *<sensor>*, está em uma determinada localização, representada pela propriedade “*Place1*”.

Os trechos de código demonstram a conformidade da linguagem proposta com o modelo IMS LD e como devem ser especificados alguns dos elementos da ML4UL. A partir dos exemplos, também é possível constatar que alguns editores existentes de IMS LD podem ser usados para especificar os novos elementos. Por fim, os trechos de codificação permitem uma análise inicial sobre qual seria o esforço para um programa interpretar essa especificação e executá-la diretamente no dispositivo móvel ou qual seria a dificuldade de conversão para um aplicativo móvel.

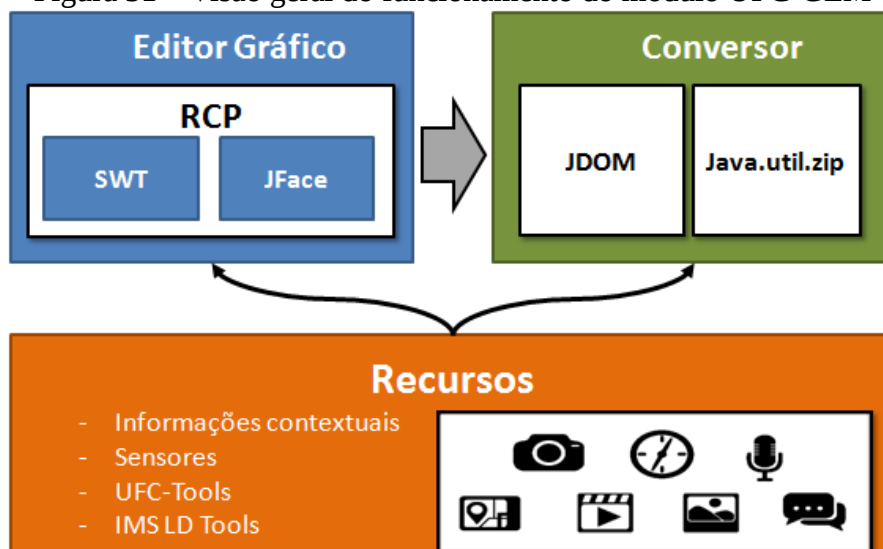
5.3 O módulo UFC-GLM

O módulo UFC-GLM trata-se de um software, para plataforma Windows, que tem como principal objetivo possibilitar que profissionais de ensino possam fazer a especificação de aulas de campo ubíquas através de recursos gráficos, como diagramas e a funcionalidade *drag & drop*. Com isso, pretende-se estabelecer uma forma simplificada para a representação das atividades de aprendizagem, do fluxo em que elas acontecem e dos recursos de computação ubíqua disponibilizados pela ferramenta que podem ser usados em campo.

5.3.1 Funcionamento

O UFC-GLM está baseado na plataforma Eclipse RCP (*Rich Client Platform*). Esta consiste em um framework que fornece um conjunto de bibliotecas e classes com o objetivo de construir ferramentas de edição, como o próprio Eclipse. A Figura 31 apresenta uma visão geral do funcionamento do UFC-GLM. Dentre as bibliotecas da RCP, o UFC-GLM utiliza a SWT (*Standard Widget Toolkit*) e a JFace UI para criação dos modelos gráficos. Para a criação, manipulação e exportação dos documentos XML é utilizada a biblioteca JDOM. O UFC-GLM ainda usa a API nativa do Java *java.util.zip* para compactar o arquivo XML e os outros documentos necessários em um arquivo único (.zip).

Figura 31 – Visão geral do funcionamento do módulo UFC-GLM



Fonte: elaborada pelo próprio autor.

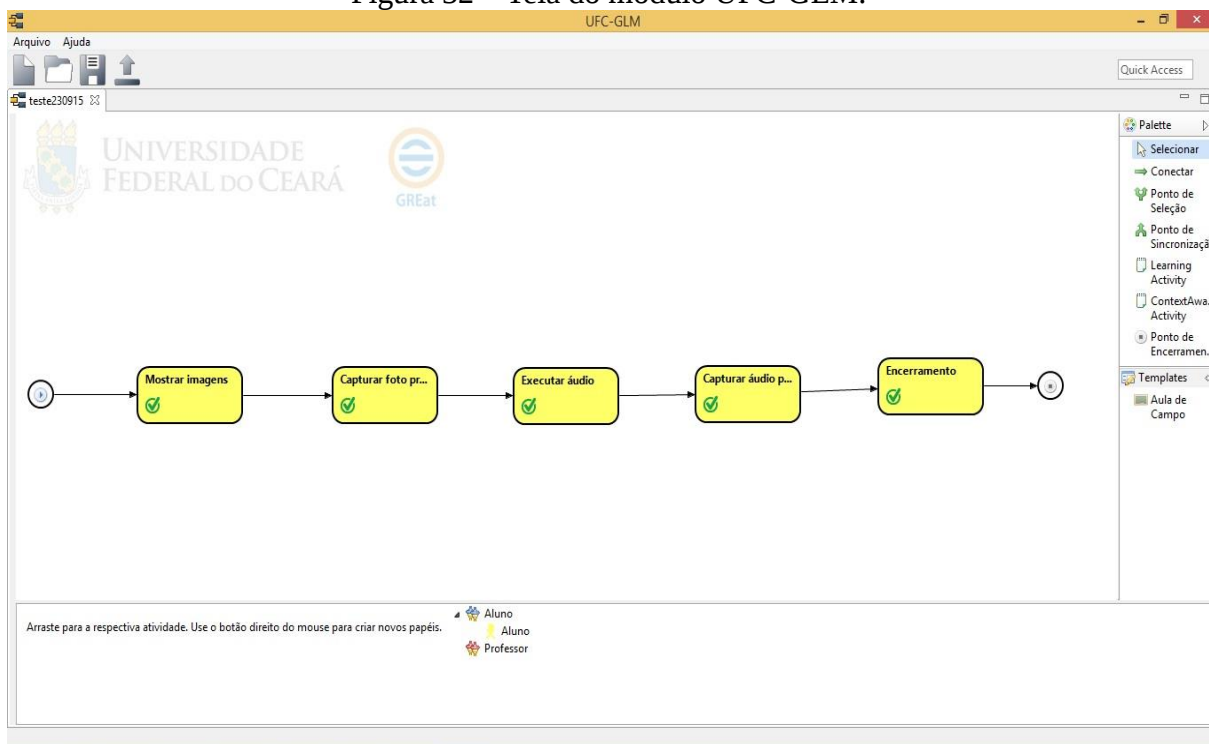
O UFC-GLM foi estendido a partir do código-fonte aberto do *Graphic Learning Modeler - GLM*, o qual é uma extensão do editor *Reload Learning Design*. Esta solução foi escolhida por ter compatibilidade com a especificação IMS LD e disponibilizar código-fonte aberto na linguagem Java (de conhecimento do autor desse trabalho).

5.3.2 Interface

A interface do UFC-GLM é dividida em quatro partes, conforme pode ser visto na Figura 32. A maior área (no centro) é usada para projetar o diagrama com a sequência das atividades de aprendizagem. Na parte superior está o menu textual e botões de acesso rápido, que têm as ações básicas da ferramenta, como sair ou abrir um novo projeto. Do lado direito, tem a paleta com os componentes para construir o diagrama da aula de campo ubíqua. Também do lado direito, estão os *templates*, que são pequenos trechos prontos de diagramas que podem ser reutilizados pelos autores das aulas de campo ubíquas.

Na parte inferior da tela, estão os símbolos que representam os participantes da unidade de aprendizagem (elemento <role>). Existem dois tipos de roles, que podem ter diferentes instâncias cada um, são eles: o Professor, que representa aquele que conduzirá as práticas educativas em campo (e.g. professor, instrutor ou monitor); e o Aluno, que corresponde aos diferentes tipos de alunos presentes naquela aula (e.g. novato ou veterano). O UFC-GLM ainda permite que vários projetos de aulas sejam abertos ao mesmo tempo, os quais podem ser acessados através das abas que ficam logo abaixo dos menus com as funções básicas.

Figura 32 – Tela do módulo UFC-GLM.



Fonte: elaborada pelo próprio autor.

Os projetos das aulas de campos são compostos essencialmente pelas atividades (*Learning-Activity* ou *ContextAware-Activity*) e as arestas que indicam a sequência das ações. Ao adicionar ou editar uma atividade, uma janela é aberta e nela é possível preencher as informações sobre aquela atividade (como descrição, objetivos e pré-requisitos) e os recursos que serão disponibilizados em campo, conforme pode ser observado na Figura 33. Através dos recursos disponíveis no UFC-GLM, o projetista poderá indicar quais funcionalidades a aplicação gerada deverá ter, as quais ficarão disponíveis para os alunos durante a aula de campo.

É importante destacar que o UFC-GLM conta com recursos criados a partir da ML4UL, porém não implementa todas as possibilidades suportadas pela linguagem, devido a mesma ter um caráter mais generalista. Por exemplo, a ML4UL conta com o elemento `<sensor>` e o UFC-GLM implementa as funcionalidades de sensoriamento GPS e bússola, entretanto não dispõe dos sensores de luminosidade e de temperatura.

- GPS. Funcionalidade que permite ao projetista especificar na aplicação a utilização do sensor GPS do dispositivo móvel para capturar os dados de localização geográfica do usuário. Ela equivale a uma implementação do elemento *<sensor>*.
- Bússola. Recurso que permite que ao projetista indicar que a aplicação deve utilizar o sensor de campo magnético do dispositivo para simular uma bússola e fornecer informações para orientação do aluno. Ela equivale a uma implementação do *<sensor>*.
- QRCode. Através desse recurso, a aplicação utilizará a câmera do dispositivo móvel para realizar a leitura de códigos QR (*Quick Response*) e fornecer informações para o aluno. Equivale a uma implementação do elemento *<sensor>*.
- Compartilhamento. Através desse recurso é possível especificar o compartilhamento de arquivos ou mensagens entre os participantes da aula de campo (alunos e professor). Equivale ao elemento *<sharing>*.
- Conferência local. Ao selecionar essa funcionalidade, o projetista informa que a aplicação móvel deverá permitir a realização de chats via rede sem fio durante a aula de campo. Ela corresponde ao elemento *<local-conference>*.

Além desses recursos, o UFC-GLM permite que o projetista adicione questões à sua aplicação, conforme mostrado na Figura 34. As perguntas podem ser subjetivas (com texto livre) ou objetivas, com número de opções e conteúdos definidos em tempo de projeto. Cada atividade pode ter um apenas um questionário, com perguntas para avaliação do aluno ou simplesmente para obtenção de informações, como nome, sexo e idade.

O UFC-GLM possibilita a especificação de atividades de aprendizagem condicionadas a regras de adaptação especificadas pelo professor em tempo de projeto. Assim, é possível representar uma Unidade de Aprendizagem adaptativa e sensível ao contexto dos alunos. Para isso, é utilizado o elemento *<contextaware-activity>* da ML4UL.

Figura 34 – Tela do UFC-GLM para criação das perguntas de uma atividade.

Questões

Definir questões

Aqui você tem a possibilidade de definir questões. Para começar, clique no botão "Nova Questão" abaixo.

Questão Nr. 1:

Resposta Nr. 1: ☐ Resposta Correta

Resposta Nr. 2: ☒ Resposta Correta

Resposta Nr. 3: ☐ Resposta Correta

Questão Nr. 2:

Fonte: elaborada pelo próprio autor.

A Figura 35 mostra a aba ‘Condições’, da tela de edição da *ContextAware Activity*, usada para configurar as regras de adaptação da atividade. O projetista poderá adicionar diferentes condições, que são montadas a partir da seleção das variáveis de contexto (elemento *<property>* criado a partir da adição dos recursos), das condicionantes (como “=”, “<” ou “>”) e dos valores a serem verificados. As condições são combinadas através das cláusulas “E” ou “OU”. Desta forma, para que uma atividade sensível ao contexto (*ContextAware Activity*) seja executada é necessário que o resultado da combinação das regras especificadas seja verdadeiro. Caso contrário, a aplicação móvel não deverá executar as ações previstas nessa atividade e passa para a próxima do fluxo da aula de campo ubíqua.

Figura 35 – Tela do UFC-GLM para configuração das regras de adaptação.

Tela de edição da Context Aware Activity

Editar Context Aware Activity
Essa tela permite você editar os parâmetros da Context Aware Activity

Nome:

☒ Context Aware Check ☒ Context Aware Activity visível

Complementos ☒ Objetivos ☒ Pré-requisitos ☒ Encerramento **Condições**

Condições

	Variável de Contexto 1	=	Valor 1
OU	Variável de Contexto 2	<>	Valor 2
E	Variável de Contexto 3	>=	Valor 3
---	---	---	
---	---	---	

Salvar

OK Cancel

Fonte: elaborada pelo próprio autor.

Após concluir a modelagem da aula de campo ubíqua, com a inclusão de todos os recursos importantes, deve-se converter o modelo gráfico para o pacote ML4UL (modelo textual mais os itens adicionados). Para isso, o projetista deve executar a função ‘Gerar pacote ML4UL’, no menu superior. Assim, o UFC-GLM interpretará o modelo gráfico criado, gerará a especificação ML4UL em XML e a compactará junto com os itens adicionados ao projeto em um arquivo ‘.zip’. Esse servirá como entrada para o módulo UFC-Generator poder criar a aplicação móvel.

5.4 O módulo UFC-Generator

O módulo UFC-Generator consiste em um sistema Web que tem como entrada o arquivo com o plano da aula de campo ubíqua especificado na linguagem ML4UL (gerado pelo UFC-GLM) e como saída a aplicação em formato nativo do sistema operacional móvel selecionado. Para dar suporte à geração de aplicações compatíveis com diferentes sistemas operacionais móveis, o UFC-Generator tem integração com a plataforma Apache CordovaTM.

5.4.1 Funcionamento

Conforme apresentado anteriormente na

Figura 26, a geração das aplicações ubíquas pelo UFC-Generator pode ser dividida em quatro fases: validação do arquivo de entrada (Validador); transformação da aula no modelo original em ML4UL para a codificação destino (Transcodificador); geração do arquivo executável da aplicação na plataforma alvo (Gerador); e, entrega da aplicação gerada para o usuário final (Publicador).

O paradigma MDD defende que erros na elaboração dos modelos serão propagados no restante do projeto, portanto a corretude dos modelos torna-se um fator chave na qualidade da aplicação final. Então, primeiramente o UFC-Generator valida o conteúdo do arquivo de entrada (.zip), verificando se o manifesto (imsmanifest.xml) contém as *tags* dos principais elementos (*activity*, *role*, *method* e *environment*) e se todos os arquivos referenciados nele estão contidos no pacote Zip. Em caso de alguma inconsistência, o processo de geração é interrompido e uma mensagem de erro é emitida ao usuário.

Após a etapa de validação do arquivo com a aula ubíqua, o UFC-Generator realiza transformações entre modelos para, a partir da especificação em ML4UL, obter a codificação na linguagem de programação destino. É importante destacar que, para fins de demonstração e considerando a escolha da plataforma CordovaTM, a linguagem utilizada foi o HTML 5 com Javascript e CSS. Entretanto, esse processo de transcodificação (codificação em ML4UL para o código-fonte da linguagem de programação alvo) poderia ser feito utilizando outra linguagem.

A

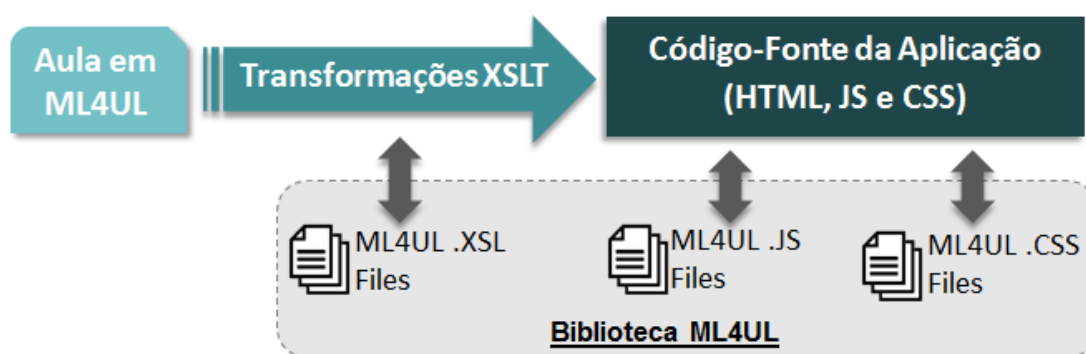
Figura 36 mostra uma visão geral do processo de transcodificação realizado pelo UFC-Generator, que, como se pode observar, é baseado em dois elementos principais: as transformações entre modelos utilizando a linguagem XSLT (*eXtensible Stylesheet Language*

Transformations) e a biblioteca de arquivos ML4UL. A XSLT é a linguagem padrão desenvolvida pelo consórcio W3C para especificação das regras de transformação entre documentos XML.

A partir do arquivo XML com a especificação da aula de campo em ML4UL são aplicadas transformações, que seguem as regras especificadas nos arquivos XSL da biblioteca ML4UL. Como resultados são gerados arquivos HTML e JS (Javascript) que especificam o funcionamento da aplicação. Os arquivos Javascript da biblioteca ML4UL contém métodos de acesso às funcionalidades da plataforma CordovaTM e irão compor o pacote para geração da aplicação final. Os arquivos CSS da biblioteca constituem os modelos disponíveis de estilos que podem ser utilizados para customização da aplicação e são referenciados no arquivo HTML.

Após o código-fonte ter sido produzido, o UFC-Generator organiza os arquivos fontes, mais os de configuração, executa o Gerador e obtém o executável da aplicação ubíqua. Nessa implementação o processo de geração é realizado pelo compilador CordovaTM. Entretanto, devido à estrutura baseada em modelos do UFC-Inventor, essa funcionalidade de geração pode ser alterada se necessário, por exemplo, no caso da descontinuidade da plataforma CordovaTM. Nessa situação, poder-se-ia optar por outra solução multiplataforma e realizar pequenos ajustes na parte da geração apenas com relação à compatibilidade das funcionalidades, sem necessitar de alterações no resto do ambiente.

Figura 36 – Visão geral do processo de transcodificação.



Fonte: Elaborada pelo próprio autor.

Esta tese de doutorado usa uma abordagem de transformação de modelos semelhante a adotada por García-Magariño e Palacios-Navarro (2016), onde a partir de modelos de alto nível são realizadas transformações até se obter o código-fonte. Entretanto, eles utilizam a linguagem ATL e geram código fonte em Java, enquanto nesse trabalho é utilizado XSLT e o resultado são arquivos HTML e Javascript. Além disso, o código gerado na proposta de

García-Magariño e Palacios-Navarro (2016) necessita ser estendido, não podendo ser compilado logo depois de gerado, diferentemente do apresentado nesta tese, cujos arquivos executáveis da aplicação são criados.

As Figuras Figura 37, Figura 38 e Figura 39 ilustram um exemplo de transformação entre modelos que pode ser realizada através do módulo UFC-Generator. Na Figura 37 é apresentado um trecho da especificação de uma aula de campo em ML4UL (*imsmanifest.xml*) gerada pelo módulo UFC-GLM, destacando-se o elemento `<ml4ul:media-capture>`. As partes de código não relacionadas com essa explicação são substituídas por “...”.

Para criar as linhas de código referentes a esse trecho de especificação ML4UL, o UFC-Generator utiliza as regras especificadas no arquivo de transformações relativo a esse elemento (*xsl_media-capture.xsl*). A Figura 38 mostra um extrato desse arquivo. É possível observar que as *tags* XSLT, como `<xsl:attribute>` e `<xsl:if>`, misturam-se com *tags* HTML, como `<p>` e `<button>`. Então, o UFC-Generator aplica todas as regras especificadas na biblioteca de arquivos ML4UL sobre a especificação da aula de campo e obtém os arquivos HTML e Javascript.

A Figura 39 mostra um trecho do arquivo HTML gerado (*index.html*), na parte referente ao elemento `<ml4ul:media-capture>`. Comparando-se esse extrato de código HTML com o trecho do arquivo ‘*xsl_media-capture.xsl*’, pode-se observar claramente equivalências entre as *tags* resultantes. Além disso, pode-se perceber que o ‘*index.html*’ gerado referencia uma função javascript (*capturarFoto()*) que faz parte do arquivo ‘*media-capture.js*’ da biblioteca de arquivos ML4UL e foi especificada no arquivo com as regras de transformação. Por fim, o UFC-Generator copia o arquivo HTML e o javascript gerados, além de outros de configuração da biblioteca ML4UL, para uma pasta específica e executa os comando de compilação CórdovaTM para gerar o arquivo executável da aplicação.

Figura 37 – Trecho de código do arquivo *imsmanifest.xml*

```

83 ...
84 <imsld:environment identifier="env-7abe1801-ddfa-1213-aeb9">
85   <imsld:title>Captura de foto</imsld:title>
86   <ml4ul:media-capture
87     identifier="lo-427ba328-ebf1-a296-9102"
88     isvisible="true"
89     tipoCaptura="foto"
90     textoSubtitulo="Capture sua própria imagem do local">
91   </ml4ul:media-capture>
92 </imsld:environment>
93 ...
94

```

Fonte: Elaborada pelo próprio autor.

Figura 38 – Trecho de código do arquivo media-capture.xsl

```

6 <xsl:stylesheet>
7 ...
8 <script type="text/javascript" charset="utf-8" src="ml4ul/ml4ulMediaCapture.js"> </script>
9 ...
10 <xsl:template match="ml4ul:media-capture">
11 <xsl:if test="@tipoCaptura = 'foto'">
12 <p> <xsl:value-of select="@textoSubtitulo" /> </p>
13 ...
14 <p> <xsl:attribute name="id"> <xsl:value-of select="textoMensagem" /> </xsl:attribute>
15 Clique no botão para capturar a foto
16 </p>
17 <button>
18 <xsl:attribute name="onclick"> <xsl:value-of select="'capturarFoto()' ' />
19 </xsl:attribute>
20 CAPTURAR FOTO
21 </button>
22 </xsl:if>
23 ...
24 </xsl:template>
25 ...
26 </xsl:stylesheet>
27

```

Fonte: Elaborada pelo próprio autor.

Figura 39 – Trecho de código do arquivo index.html

```

2 <html>
3 <script type="text/javascript" charset="utf-8" src="ml4ul/ml4ulMediaCapture.js"></script>
4 ...
5 <body>
6 ...
7 <p> Capture sua própria imagem do local </p>
8 ...
9 <p> Clique no botão para capturar a foto </p>
10 ...
11 <button onclick="capturarFoto()"> CAPTURAR FOTO </button>
12 ...
13 </body>
14 </html>
15

```

Fonte: Elaborada pelo próprio autor.

Para disponibilizar as aplicações geradas para os usuários finais, o UFC-Generator apresenta duas opções. Após o executável ser criado, um link direto para o arquivo é disponibilizado. Depois, é possível enviar o arquivo com a aplicação para os e-mails dos usuários destinatários, por exemplo, os alunos de uma aula de campo.

5.4.2 Interface

A Figura 40 apresenta a tela do sistema UFC-Generator. Percebe-se que a interface é simples e com poucas funções, para facilitar o uso. Essencialmente, o usuário deve executar os seguintes passos: selecionar o arquivo (.zip) com o projeto da aula de campo em ML4UL (botão ‘Escolher arquivo’); escolher o sistema operacional móvel no qual a aplicação será gerada (Android, iOS ou Windows Phone); optar por um dos temas de aplicação disponíveis;

e, clicar no botão ‘Gerar App’. O link para o arquivo gerado será disponibilizado ao lado do botão ‘Gerar App’.

Figura 40 – Tela do sistema UFC-Generator.

Fonte: elaborada pelo próprio autor.

5.5 Conclusão

Esse capítulo apresentou o UFC-Inventor, com o detalhamento da linguagem ML4UL, que consiste no núcleo do ambiente, e os dois módulos UFC-GLM e UFC-Generator. Com relação à especificação da ML4UL, foram descritos todos os recursos e elementos disponíveis, além da apresentação de um cenário de aplicação e de exemplos de usos. Com relação aos dois módulos, para ambos foram detalhadas as formas de funcionamento dos processos e as funcionalidades das interfaces. No próximo capítulo será apresentada a avaliação do UFC-Inventor, de forma a demonstrar o processo completo relacionado ao uso de todos os componentes do ambiente.

6. AVALIAÇÃO DO AMBIENTE UFC-INVENTOR

Este capítulo descreve as atividades realizadas para avaliação do ambiente UFC-Inventor. A seção 6.1 apresenta a metodologia utilizada. Na seção 6.2 são descritos como foram elaborados os instrumentos aplicados. A seção 6.3 descreve como se deu a avaliação da usabilidade com os professores. Na seção 6.4 é descrito o estudo de caso que foi realizado para comprovação do processo completo do UFC-Inventor, desde o projeto e geração da aplicação ubíqua, passando pelo seu uso em uma aula de campo real, até a visualização dos dados coletados e transmitidos pelos alunos. Por fim, a Seção 6.5 conclui este capítulo.

6.1 Metodologia

Segundo Ainsworth e Grimshaw (2004), a análise plena de uma ferramenta de autoria depende da observação dos resultados tanto do software de geração quanto das aplicações geradas. Considerando isso, a avaliação do UFC-Inventor foi realizada em duas etapas: verificação da usabilidade do ambiente com professores e realização de um estudo de caso com alunos em uma aula de campo real.

Um aspecto que reforça a importância do teste em campo é o fato de pesquisas indicarem que diferentes questões ambientais podem dificultar o uso dos dispositivos móveis em atividades de campo (MARÇAL *et al.*, 2014; MEEK *et al.*, 2013; CHEN; LIU, HWANG, 2015; ŘEŘUCHA *et al.*, 2015), por exemplo: excesso de iluminação ou chuvas, que dificultam a visualização da tela do dispositivo tanto para escrita de texto quanto para captura de fotos; ausência de conexão com a Internet, que impossibilita a troca de informações remotas; queda da bateria do dispositivo e indisponibilidade de meios para carregamento imediato; entre outros.

Através do teste de usabilidade do UFC-Inventor, pretendeu-se verificar se os professores conseguiriam modelar uma aula de campo ubíqua e gerar as respectivas aplicações através do ambiente. Com relação ao estudo de caso em campo, o objetivo principal foi avaliar o desempenho do ambiente em condições normais de uso, diferentemente de testes controlados em laboratório ou dentro da própria universidade. Com isso, buscou-se verificar as seguintes questões: se a proposta de transformações entre modelos para obter aplicações ubíquas multiplataformas a partir de representações de alto nível funcionava; se um aplicativo gerado pelo UFC-Inventor apresentava bons níveis de usabilidade, capacidade de aprendizado e utilidade para os alunos; e, se o professor aprovava a adoção do ambiente

UFC-Inventor como ferramenta de apoio às aulas de campo, do pré ao pós-campo.

6.2 Instrumentos

Existem disponíveis na literatura diferentes questionários padronizados para avaliar as respostas dos participantes de um estudo e aferir o nível de diferentes características de um sistema. As principais vantagens da utilização desses instrumentos para avaliação são: objetividade na coleta das informações, replicabilidade do instrumento em outros estudos e quantificação dos resultados a partir das respostas dos participantes, através de cálculos estatísticos (SAURO; LEWIS, 2012).

Para avaliação dos resultados desta tese de doutorado foram desenvolvidos dois questionários padronizados (os instrumentos completos se encontram no APÊNDICE B desta tese) que se baseiam em outros existentes: um instrumento para avaliação da usabilidade do UFC-Inventor, composto por duas partes, e outro para o estudo de caso em campo com os alunos, dividido em três partes.

O instrumento para avaliação da usabilidade do UFC-Inventor foi elaborado baseado no questionário *System Usability Scale* - SUS (BROOKE, 1996), do qual foram extraídas as dez primeiras questões. Além dessas, foi adicionada uma questão aberta para o usuário indicar os pontos positivos, negativos e sugestões de melhorias para o ambiente.

O questionário para avaliação com os alunos no estudo de caso em campo é mais completo, conta com dezoito perguntas e é dividido em três partes: a primeira parte, baseada no SUS, objetiva coletar informações sobre a facilidade de uso (Usabilidade) da aplicação gerada e a simplicidade para aprender a usá-la (Capacidade de Aprendizado); a segunda parte, baseada no Modelo de Aceitação de Tecnologia de Davis (DAVIS, 1989), é voltada para identificar o nível de utilidade do aplicativo percebido pelos alunos (Utilidade Percebida) durante o uso em campo; a terceira e última parte do questionário é composta por duas questões abertas, que objetivam coletar as opiniões dos alunos sobre preferências na utilização de recursos da aplicação, pontos positivos, negativos e sugestões de melhorias.

A escala SUS se caracteriza como um método de fácil aplicação para averiguação da usabilidade de sistemas. Diversos estudos demonstram o uso do questionário SUS para avaliar aplicações móveis educativas e ferramentas de autoria (ZBICK et al., 2015; ALAMER; AL-OTAIBI; AL-KHALIFA, 2015; TABUENCA et al., 2014; SCHMITZ et al., 2013). O questionário da escala SUS é composto por 10 questões (ou itens), cada uma com cinco opções de respostas que seguem a escala Likert de 5 pontos: de 1 (discordo plenamente) a 5

(concordo plenamente), em que 3 significa neutro.

Para calcular a pontuação referente à primeira parte do instrumento de avaliação, são somadas as contribuições de cada item de pontuação. Para as questões redigidas positivamente (1, 3, 5, 7 e 9), a contribuição da pontuação é o valor da resposta menos 1. Para as questões redigidas negativamente (2, 4, 6, 8 e 10), a contribuição é igual a 5 menos o valor da resposta. Posteriormente, somam-se os valores das contribuições das 10 questões e multiplica-se por 2,5 para se obter o valor do escore SUS. Conforme afirmado por Lewis e Sauro (2009), o SUS foca na análise de dois fatores principais do sistema: usabilidade, compreendido pelas questões 1, 2, 3, 5, 6, 7, 8 e 9; e, capacidade de aprendizado, compreendido pelas questões 4 e 10.

Seguem abaixo as 10 questões que compõem a primeira parte dos instrumentos de avaliação desta tese e que são baseadas no questionário SUS.

- 1) Eu acho que gostaria de usar essa aplicação frequentemente.
- 2) Eu achei essa aplicação desnecessariamente complexa.
- 3) Eu achei a aplicação fácil para usar.
- 4) Eu acho que precisaria do apoio de um suporte técnico para usar essa aplicação.
- 5) Eu achei que as várias funções da aplicação estavam bem integradas.
- 6) Eu achei que havia muita inconsistência na aplicação.
- 7) Imagino que a maioria das pessoas possa aprender a utilizar este aplicativo muito rapidamente.
- 8) Achei a aplicação muito complicada de se usar.
- 9) Eu me senti muito confiante em utilizar esta aplicação.
- 10) Eu precisei aprender várias coisas antes que eu pudesse começar a usar essa aplicação.

Segundo Davis (1989), um sistema útil é aquele que ajuda o usuário a melhorar o seu desempenho na execução de um trabalho. Para ele, a usabilidade é um fator importante, porém se o usuário não perceber a utilidade do sistema, não irá utilizá-lo. Davis propôs um modelo que permite a quantificação do grau de utilidade percebida pelos usuários de uma determinada aplicação: o Modelo de Aceitação de Tecnologia (em inglês Davis' Technology Acceptance Model – TAM). Diversos estudos têm utilizado o modelo de Davis para avaliação da aceitação de sistemas (MUK; CHUNG, 2015; CHANG; HUANG, 2015; RATNA; MEHRA, 2015; AKMAN; TURHAN, 2015).

As questões que compõem a segunda parte do instrumento de avaliação com os alunos

são baseadas no modelo TAM, com adaptações para o contexto de aplicações ubíquas, semelhantemente aos estudos de Zbick et al. (2015) e Chung, Chen e Kuo (2015). Nesta parte do instrumento também é adotada a escala Likert. As questões referem-se à aplicação móvel e são apresentadas a seguir.

- 11) Facilitou-me coletar os dados usando recursos como a câmera e o microfone.
- 12) Dificultou-me a escrever textos e números.
- 13) Permitiu-me executar as tarefas em campo mais eficientemente.
- 14) Complicou a minha execução das atividades de aprendizagem.
- 15) É uma tecnologia útil para coleta dos dados.
- 16) Ajudou-me a compreender melhor os conceitos da aula de campo.

6.3 Análise da usabilidade do UFC-Inventor

Para obter a percepção dos professores sobre o ambiente, foi realizada uma análise específica sobre a usabilidade dos módulos UFC-GLM e UFC-Generator. Esse processo de avaliação está ilustrado na Figura 41 e ocorreu sequencialmente da seguinte forma: inicialmente, uma entrevista para identificar o perfil dos participantes; posteriormente, um treinamento para demonstrar o funcionamento e as funcionalidades dos módulos do UFC-Inventor; depois, três tarefas que os participantes deveriam realizar usando o ambiente; e, por fim, um questionário para avaliar o UFC-Inventor a ser respondido pelos professores.

É importante destacar que essa avaliação não aconteceu em um único momento com todos os participantes. Ela ocorreu de forma individualizada, de acordo com a disponibilidade de tempo de cada um deles. A entrevista era simples e composta apenas das seguintes perguntas: qual a idade, em que área de ensino atuava, quais os conhecimentos e experiência de informática que detinha e há quanto tempo era professor.

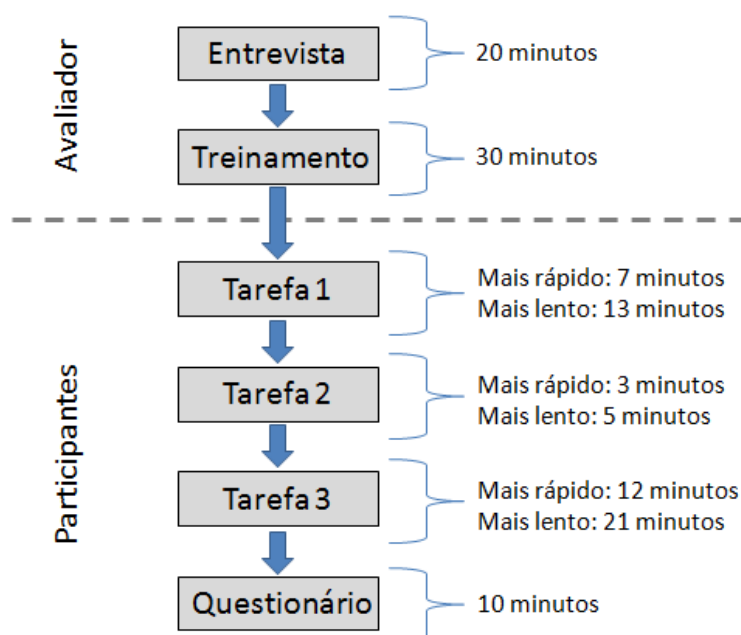
Participaram dessa avaliação cinco professores, com faixa etária entre 35 e 49 anos, que aplicam (ou podem aplicar) aulas de campo em suas disciplinas: o professor do curso de graduação em Geologia da Universidade Federal do Ceará (UFC); um professor do curso de graduação em Zootecnia da UFC; e três professores do ensino fundamental, sendo dois de Geografia e um de Biologia. Todos eles tinham prática com o uso de software em desktop, como editores de texto. Com relação ao tempo de sala de aula, a experiência deles variava entre seis e quinze anos de docência.

Foi realizado um treinamento com os professores, que objetivava capacitá-los a modelar uma aula de campo e gerar a aplicação correspondente utilizando o UFC-Inventor.

Foi explicado aos professores como a ferramenta funcionava, suas principais funcionalidades e o passo a passo para eles poderem criar suas próprias aplicações. Após essa preparação inicial, foi solicitado que eles realizassem as seguintes tarefas usando as ferramentas do ambiente: na primeira tarefa, o participante deveria usar o módulo UFC-GLM para modelar uma aula de campo com os recursos de imagem e áudio; a segunda tarefa consistia em usar o módulo UFC-Generator para gerar a aplicação modelada anteriormente; na terceira fase, os participantes deveriam realizar o processo completo, modelando uma aula com imagem, áudio e captura de foto usando o UFC-GLM, e depois, gerar a aplicação através do UFC-Generator. Através dessas tarefas, buscou-se promover a aprendizagem gradual do UFC-Inventor por meio de atividades práticas simples com diferentes recursos do ambiente.

A Figura 41 mostra o desempenho dos participantes com relação ao tempo para execução das tarefas. A primeira informação importante e animadora é o fato dos cinco participantes terem conseguido cumprir as três tarefas, mesmo havendo diferenças consideráveis nos tempos realizados. Aquela executada mais rapidamente foi a segunda tarefa, onde o participante mais lento levou cinco minutos para executá-la. A terceira foi a mais demorada e que apresentou a maior variação de tempo entre o participante mais rápido e o mais lento. Isso pode ser justificado pelo fato dela ter um nível maior de dificuldade e envolver o uso dos dois módulos do ambiente.

Figura 41 – Processo para avaliação da usabilidade do UFC-Inventor.



Fonte: elaborada pelo próprio autor.

A Tabela 11 mostra o valor do escore SUS para o UFC-Inventor obtido nessa avaliação e a classificação dele de acordo com dois estudos diferentes. Conforme pode se observar, utilizando a categorização de Bangor, Kortum e Miller (2009), o ambiente UFC-Inventor enquadra-se na categoria C, que corresponde nessa escala ao nível “Bom” de usabilidade. De acordo com a classificação de Sauro e Lewis (2012), o UFC-Inventor obteve a nota A-, que equivale ao terceiro melhor grau de usabilidade, segundo essa categorização. Pode-se observar que o UFC-Inventor alcançou um bom nível de usabilidade, considerando tanto as classificações apresentadas quanto o fato de ter sido superior ao escore mínimo aceitável de usabilidade, que é 70 segundo Bangor, Kortum e Miller (2009).

Tabela 11 – Classificação da usabilidade do aplicativo conforme duas escalas diferentes.

Sistema	Escore SUS	Classificação Bangor, Kortum e Miller		Classificação Sauro e Lewis	
		Faixa	Nota	Faixa	Nota
UFC-Inventor	79,0	70 – 80	C (Bom)	78,9 – 80,7	A-

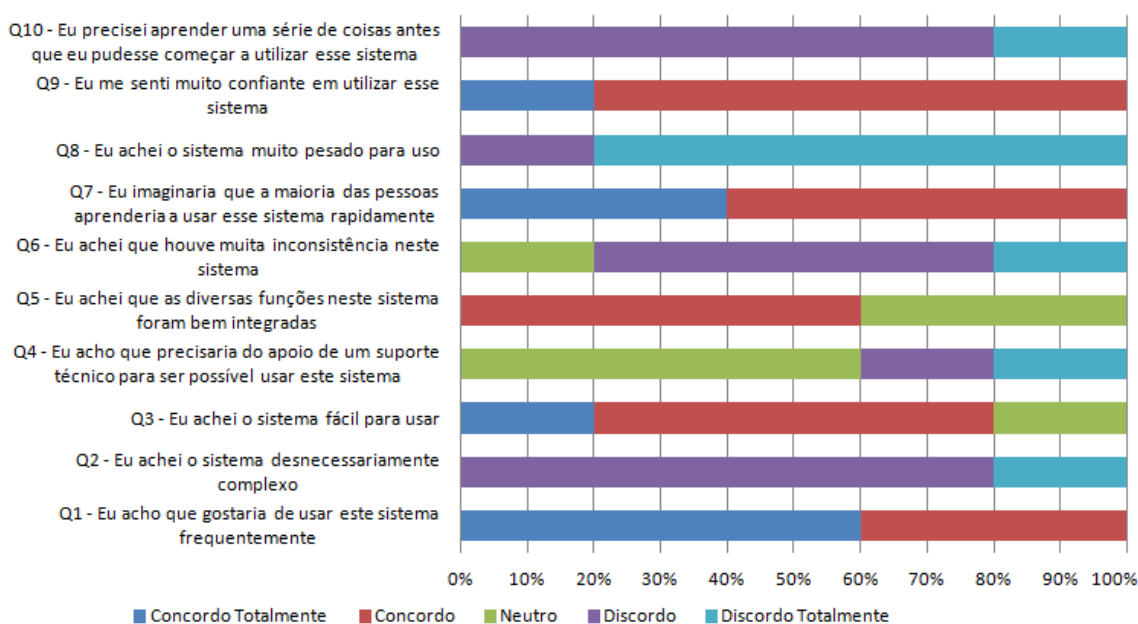
Fonte: Elaborada pelo próprio autor.

A Figura 42 apresenta a frequência (%) das respostas dos participantes e demonstra os resultados positivos com relação à avaliação da usabilidade do UFC-Inventor. Em três itens (Q1, Q7 e Q9), todos os participantes responderam afirmativamente (“Concordo” ou “Concordo Totalmente”) que pretendem continuar utilizando o sistema, que acreditam ser fácil aprender a usá-lo e que se sentiram confiantes durante a utilização. Das questões redigidas positivamente, a que teve as piores avaliações foi a Q5 (obteve média na avaliação de 3,6), que diz respeito à integração das funções do ambiente. Das questões redigidas negativamente, a que teve as piores avaliações foi a Q4 (obteve média na avaliação de 2,4), que diz respeito à necessidade de suporte para uso do sistema. A partir desses valores e das observações realizadas pelo avaliador durante o teste, acredita-se que esses resultados estão relacionados ao fato do UFC-Inventor ser composto por dois módulos diferentes com plataformas e interfaces distintas. De uma forma geral, os resultados médios das avaliações são favoráveis: média dos itens redigidos positivamente igual a 4,16; e média dos itens redigidos negativamente igual a 1,84.

Com relação aos aspectos positivos indicados pelos participantes na pergunta aberta, destacam-se: a capacidade do UFC-Inventor de permitir a criação de aplicações com conteúdos diferentes, a não obrigatoriedade de o usuário ter conhecimentos de programação para usar o ambiente e a possibilidade das aplicações geradas funcionarem em diferentes sistemas operacionais, como Android e Windows Phone. Duas críticas foram apontadas pelos

participantes sobre o UFC-Inventor. A primeira foi a impossibilidade de se visualizar como a aplicação está ficando durante o Projeto, que pode ser contornada através de gerações parciais da aplicação. O segundo ponto negativo indicado foi a ausência de um manual que descrevesse passo a passo como executar as funcionalidades do ambiente.

Figura 42 – Frequência (%) das respostas da segunda parte do instrumento de avaliação.



Fonte: Elaborada pelo próprio autor.

6.4 Estudo de caso

Esta seção descreve como se deu avaliação do ambiente UFC-Inventor considerando o processo completo, desde a modelagem através do UFC-GLM, passando pela geração da aplicação, até o uso dela pelos alunos em uma aula de campo real.

6.4.1 Participantes

Para a realização do estudo de caso foi escolhida a disciplina “Mapeamento de Terrenos Sedimentares” do curso de graduação em Geologia da Universidade Federal do Ceará. Esta escolha se deu devido aos seguintes fatores: essa disciplina ter um grande número de aulas de campo; a agenda dessas visitas ao campo encaixava com a conclusão da implementação do UFC Inventor; e, o professor dessa disciplina ter experiência no uso de aplicativos móveis em suas aulas. Esta disciplina tem como objetivo principal treinar os alunos em técnicas de mapeamento geológico de bacias sedimentares utilizando aulas práticas de campo. O estudo foi realizado no semestre letivo 2015.2, em dezembro de 2015, na região

da Bacia do Araripe que fica no município de Nova Olinda, no interior do estado do Ceará – Brasil, a 504 quilômetros de distância da capital Fortaleza. Nessa aula de campo, os alunos deveriam visitar seis afloramentos geológicos para alcançar os objetivos de aprendizagem da disciplina. Durante o uso do aplicativo nos locais dos afloramentos, observou-se que não havia conexão de Internet móvel e, por vezes, não havia nenhum tipo de sinal nos telefones celulares.

A turma selecionada era composta por um professor e dezessete alunos, com faixa etária entre 20 e 27 anos. O professor já havia participado de outra experiência com o uso de aplicativos no ensino de Geologia (MARÇAL et al., 2015), porém nunca havia utilizado uma ferramenta de autoria de aplicações educativas. Com relação aos alunos, todos já tinham experiência com aulas de campo, devido ao semestre avançado da disciplina escolhida (realizada no penúltimo ano do curso). Entretanto, nenhum deles havia utilizado outro aplicativo educativo em campo anteriormente. Sobre os dispositivos móveis deles, apenas quatro alunos tinham celulares que não suportavam a aplicação, ou seja, que tinha sistema operacional incompatível com o aplicativo gerado. Celulares extras foram levados para contornar esse tipo de problema e possibilitar que todos os alunos utilizassem dispositivos de forma individualizada para realizarem os testes com o aplicativo gerado. Dos dezessete dispositivos utilizados, quinze equipamentos tinham o sistema operacional Android e dois utilizavam Windows Phone. É importante destacar que esses sistemas operacionais representam dois dos três mais usados em dispositivos móveis. O UFC-Inventor também gera aplicações para o sistema iOS, porém não foi gerada uma versão do estudo de caso, pois o processo para disponibilizar os aplicativos para uso nessa plataforma envolvia autorizações de terceiros, que poderiam atrapalhar o estudo de caso.

6.4.2. Procedimento

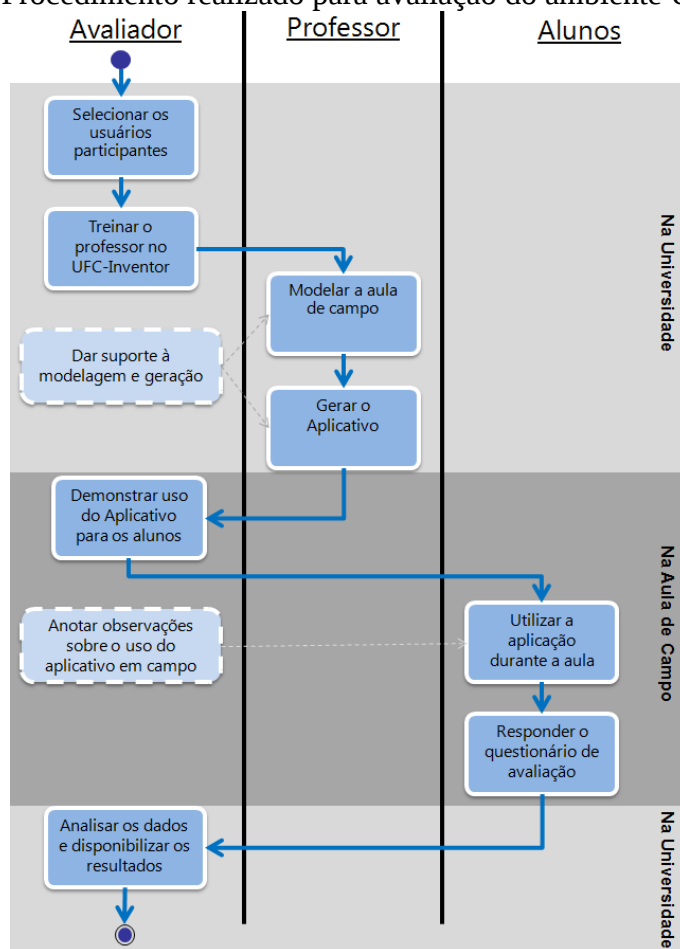
Para realizar uma análise completa do UFC-Inventor, o processo de avaliação seguiu as fases que compõem uma aula de campo (SHAKIL; FAIZI; HAFEEZ; 2011): desde o planejamento (pré-campo), passando pela aula de campo em si, até a consulta dos dados após o retorno da aula (pós-campo).

A Figura 43 apresenta um diagrama com todas as ações executadas. Pode-se observar a participação de três atores: o avaliador (autor desta tese de doutorado), o professor da disciplina, e os alunos. O processo se inicia com a seleção da turma (professor e alunos) que vai participar da avaliação. Além disso, através da figura percebe-se que a avaliação ocorreu

em dois espaços: na Universidade, nas fases de pré e pós-campo executadas pelo avaliador e pelo professor; e no campo, enquanto os alunos participavam da aula.

Após a definição dos participantes, o avaliador realizou um treinamento para o professor sobre a utilização do UFC-Inventor, para que ele pudesse modelar a aula de campo e gerar o aplicativo para os dispositivos móveis. Foram repassadas as explicações necessárias para o professor poder criar um projeto novo, modelar uma aula de campo com os recursos de computação móvel e ubíqua e gerar a aplicação relativa ao projeto elaborado. Posteriormente, o avaliador acompanhou o uso do ambiente pelo professor, dando suporte quando necessário, tanto na modelagem da aula de campo ubíqua quanto na geração da aplicação.

Figura 43 – Procedimento realizado para avaliação do ambiente UFC-Inventor



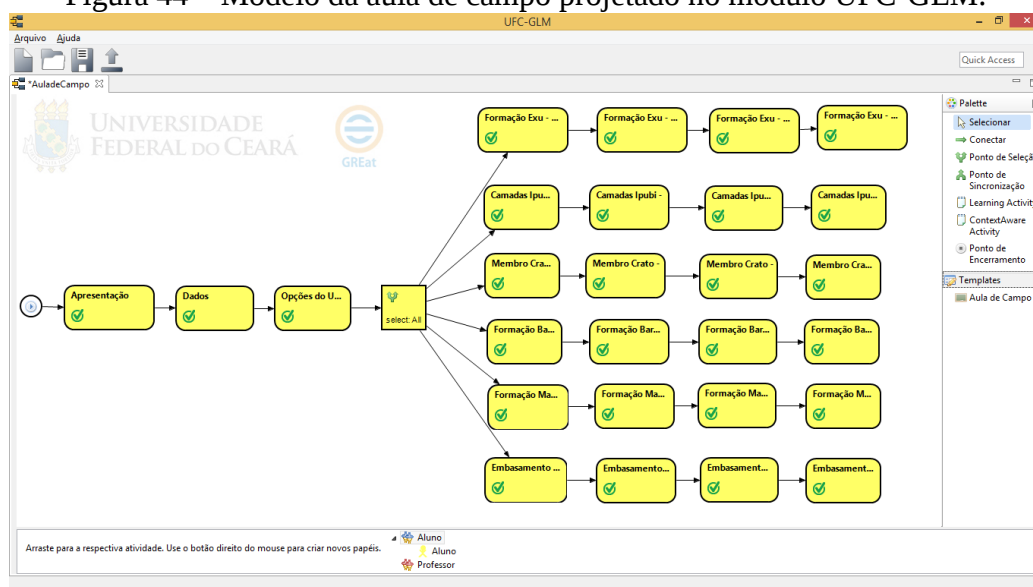
Fonte: elaborada pelo autor.

A Figura 44 apresenta o diagrama final da aula de campo modelado pelo professor. Pode-se observar que primeiramente são realizadas três atividades iniciais, até chegar a um ponto de escolha para o aluno, onde ele poderá selecionar atividades que irão variar de acordo com o afloramento geológico. Nesse modelo de aula, o estudo de cada um dos afloramentos é composto pela mesma quantidade de atividades (quatro), utilizando os mesmos recursos: explicações textuais, fotos das regiões estudadas, perguntas a serem respondidas e captura de

áudio, de fotos e da localização dos alunos (via GPS). Entretanto, é importante destacar que a ferramenta não restringe a esse estilo de aula, ficando a definição do modelo de aula a cargo do professor.

A partir do modelo da aula de campo elaborado pelo professor, foram geradas versões da aplicação compatíveis com os dois sistemas operacionais dos dispositivos móveis dos alunos (Android e Windows Phone). O aplicativo gerado é composto por 52 (cinquenta e duas) telas com textos instrutivos, imagens explicativas e perguntas a serem respondidas, além dos recursos de câmera, microfone e GPS do dispositivo móvel.

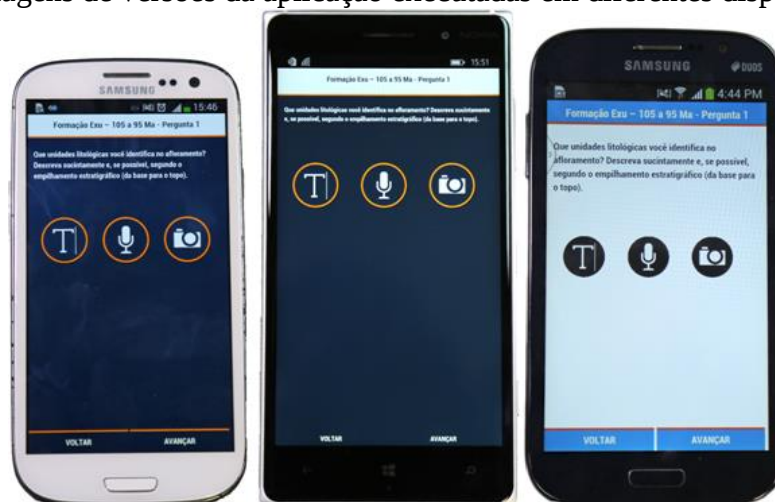
Figura 44 – Modelo da aula de campo projetado no módulo UFC-GLM.



Fonte: elaborada pelo próprio autor.

A Figura 45 mostra três variações da aplicação obtidas a partir do mesmo modelo: a primeira imagem mostra o aplicativo em um *smartphone* com sistema Android; na segunda, o aplicativo sendo executado em um dispositivo com Windows Phone; e na terceira imagem, em outro dispositivo Android, o aplicativo é apresentado com outro estilo visual.

Figura 45 – Imagens de versões da aplicação executadas em diferentes dispositivos móveis.



Fonte: elaborada pelo autor.

Antes de iniciarem a aula de campo, o avaliador demonstrou o funcionamento do aplicativo gerado aos alunos. Após a demonstração da aplicação, a aula de campo iniciou-se, com as explicações teóricas e as definições das atividades geológicas a serem realizadas. Durante a aula de campo, foram visitados seis afloramentos geológicos, das 8h da manhã até às 3h da tarde.

A Figura 46 mostra fotos com momentos do teste em campo. Nesse período, os alunos escutavam as explicações do professor, realizavam anotações na caderneta de campo e utilizavam o aplicativo, para consultar conteúdos e responder as perguntas contidas nele. A figura exhibe os alunos utilizando o aplicativo em dois afloramentos distintos, em momentos diferentes da aula de campo.

Figura 46 – Uso do aplicativo pelos alunos em diferentes momentos da aula de campo.



Fonte: elaborada pelo autor.

É importante destacar que o teste foi realizado de forma a não atrapalhar o andamento normal da aula de geologia, mantendo-se os mesmos locais e horários em que ocorrem normalmente. No decorrer da aula, o avaliador registrava observações sobre o uso do aplicativo e outros aspectos relacionados à utilização dos dispositivos móveis. Após a visita a todos os afloramentos geológicos previstos naquele dia, os alunos responderam o instrumento de avaliação, ainda na região da aula de campo.

Por fim, após o retorno para Universidade, o avaliador realizou a análise dos dados, considerando as respostas dos alunos aos questionários de avaliação, as observações anotadas em campo e os dados registrados pelos alunos através do aplicativo. O detalhamento da análise dos resultados é apresentado na próxima seção.

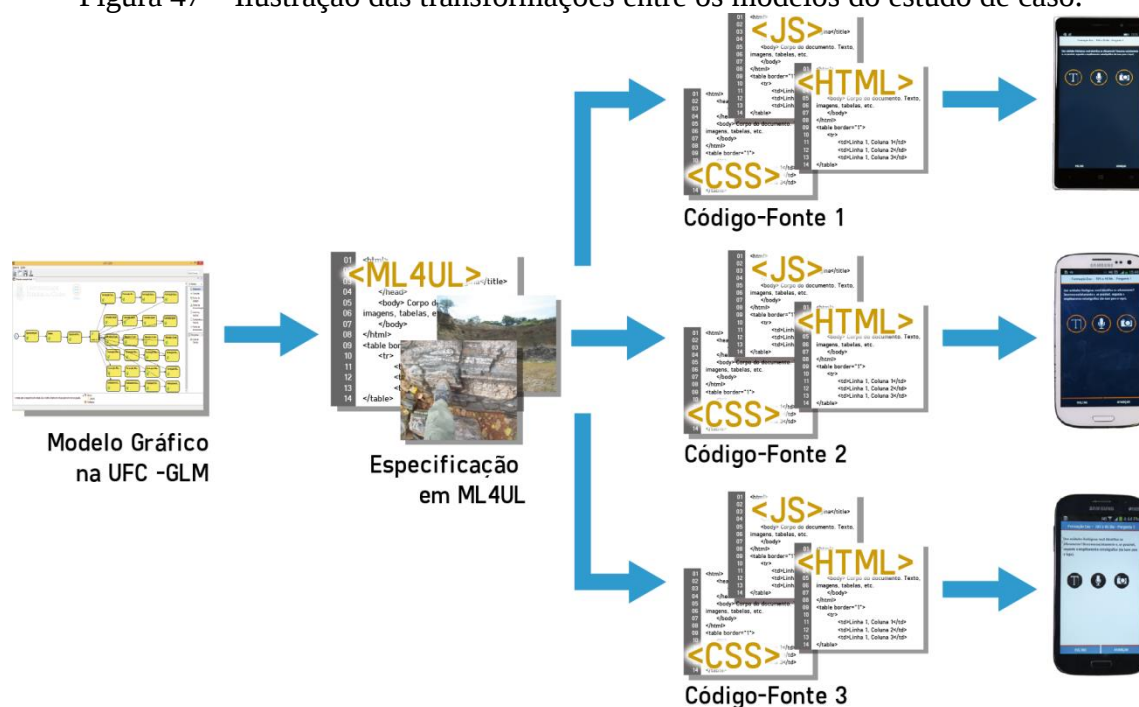
6.4.3. Resultados

Esta seção apresenta os resultados do estudo de caso em campo. Inicialmente é descrito o funcionamento completo do processo, desde a modelagem gráfica feita pelo professor até a geração das aplicações multiplataforma, passando pela produção da especificação da aula de campo em ML4UL. Posteriormente, são detalhadas as constatações sobre o uso feito pelos alunos participantes do estudo, ressaltando-se os níveis de usabilidade, capacidade de aprendizado e utilidade percebida da aplicação. Depois, são apresentados exemplos dos dados coletados pelos alunos através do aplicativo gerado. Por fim, são descritas as principais observações do professor sobre o teste com o UFC-Inventor em uma aula real.

6.4.3.1. Funcionamento do Processo Completo

O primeiro resultado importante obtido através do estudo de caso com o UFC-Inventor foi a avaliação do processo completo proposto pelo ambiente, de modo que a aplicação projetada pelo professor pôde efetivamente ser usada pelos alunos em uma aula de campo real. A Figura 47 ilustra as transformações entre os modelos instanciados para o estudo de caso do UFC-Inventor. É apresentada a geração de uma especificação única na linguagem ML4UL, a partir do modelo gráfico elaborado, e de três diferentes códigos-fonte, com base na codificação ML4UL.

Figura 47 – Ilustração das transformações entre os modelos do estudo de caso.



Fonte: elaborada pelo próprio autor.

É importante destacar que, após elaborar o modelo da aula de campo, o professor de geologia conseguiu gerar, sem intervenção de programadores, a aplicação em três versões diferentes: duas com estilos distintos para Android e uma para Windows Phone. Estas foram usadas durante o estudo de caso.

Entretanto, ainda durante o desenvolvimento do UFC-Inventor e depois, na instalação da aplicação nos dispositivos móveis, observações importantes sobre o funcionamento do ambiente foram constatadas. São elas:

1ª) Observou-se que o UFC-Inventor deveria gerar códigos-fonte diferentes para versões distintas. Inicialmente, imaginava-se que a partir do modelo em ML4UL, poder-se-ia gerar um único código-fonte e, com base nele, gerar-se-ia os diferentes arquivos executáveis de acordo com o sistema operacional alvo. Entretanto, as variações na implementação da plataforma CordovaTM nos diferentes sistemas operacionais impediu isso. Por exemplo, a codificação para persistência dos dados implementada para Android não funcionou para Windows Phone, porque as API de banco de dados eram diferentes. Nesse caso, foi necessário especificar as variações de implementação nas regras de transformação XSLT, para permitir a geração de código-fonte próprio de cada SO.

2ª) Além do código-fonte em si, constatou-se que os sistemas operacionais demandavam versões específicas das folhas de estilo (arquivos CSS). Durante os testes de programação, a partir da aplicação do estudo de caso, pode-se observar diferenças e falhas no

layout. Por exemplo, a mesma interface que funcionava normalmente no Android, ficava com problema de rolagem e os componentes desapareciam no Windows Phone. Foi necessária a construção de arquivos CSS específicos para cada sistema operacional.

3ª) Durante os testes de implantação das aplicações nos dispositivos, verificou-se que sistemas operacionais diferentes demanda ações de implantação distintas. No caso do Android, o arquivo executável (.apk) gerado pôde ser instalado diretamente no dispositivo móvel. Diferentemente do Android, com o Windows Phone houve a necessidade de se desbloquear os celulares, o que foi feito usando ferramentas da própria Microsoft. Entre os dispositivos móveis do estudo, não havia aparelhos com sistema operacional iOS, mas espera-se que essa plataforma também demande um outro processo específico para a instalação de aplicações.

Durante o uso da aplicação em campo, os recursos nas diferentes versões tiveram funcionamento semelhante, funcionando satisfatoriamente. Por exemplo, independentemente do sistema operacional ou do layout utilizado, a navegação entre as telas seguia a mesma sequência. Todas as imagens e textos foram exibidos nos seus devidos lugares. Todos os dados anotados pelos alunos foram persistidos e os recursos de captura de áudio, de foto e da localização funcionaram da forma como foram projetados.

As conclusões apontadas nesta seção estão mais relacionadas ao funcionamento completo das transformações no estudo de caso. Na próxima seção, o foco está na análise específica sobre como se deu o uso da aplicação gerada em uma aula de campo real e quais as percepções dos alunos sobre ela.

6.4.3.2. Avaliação da Aplicação Gerada

Esta seção descreve os resultados da análise do uso pelos alunos do aplicativo gerado, com base nas respostas deles ao instrumento de avaliação. Pretende-se verificar especificamente os fatores associados à usabilidade, à capacidade de aprendizado e a utilidade percebida pelos participantes do estudo de caso sobre a aplicação testada.

A Tabela 12 apresenta um resumo da análise sobre a primeira parte do questionário de avaliação, que corresponde às questões baseadas na escala SUS. Os resultados demonstram que a aplicação recebeu uma boa avaliação de usabilidade (83,4), conforme indicadores apresentados ao longo dessa seção. Além disso, também pode se afirmar, com 95% de confiança, que o escore SUS para essa população está entre 79,1 e 87,6. Os cálculos estatísticos realizados para obtenção dos valores mostrados na Tabela 12 foram baseados nas

orientações do livro de Sauro sobre as melhores práticas para uso do SUS (SAURO;2011).

Tabela 12 – Resumo da análise da primeira parte do questionário de avaliação.

Variável	Valor
Tamanho da Amostra	17
Escore Médio SUS	83,4
Intervalo de Confiança	79,1 – 87,6
Margem de Erro	4,2
Nível de Confiança	95%
Desvio Padrão	8,2
Confiabilidade	0,819

Fonte: elaborada pelo próprio autor.

Para atestar a confiabilidade dos dados obtidos, utilizou-se o coeficiente alfa de Cronbach (BONETT; WRIGHT, 2015). Ele mede a correlação em um questionário através da análise das respostas dadas pelos respondentes, apresentando uma relação média entre as perguntas. O maior valor possível para esse coeficiente é 1,00, sendo 0,70 considerado o limite inferior para uma confiabilidade interna aceitável (SAURO, 2011). Como pode se observar na Tabela 12, o coeficiente alfa de Cronbach obtido em nosso estudo foi de 0,819, caracterizando a amostra com um bom nível de confiabilidade.

Considerando-se o escore médio SUS, alguns estudos apresentam classificações baseadas nesse parâmetro para categorizar o nível de usabilidade de um sistema. A

Tabela 13 mostra como o aplicativo testado se classifica com relação a duas escalas diferentes. Conforme pode se observar, utilizando a categorização de Bangor, Kortum e Miller (2009), a aplicação “Aula de Campo” enquadra-se na categoria B, que corresponde ao nível “Excelente” de usabilidade nessa escala. Sauro e Lewis (2012) apresentam uma escala na qual o aplicativo testado obtém nota A, que se trata do segundo melhor nível de classificação de usabilidade segundo essa categorização, abaixo apenas do A+.

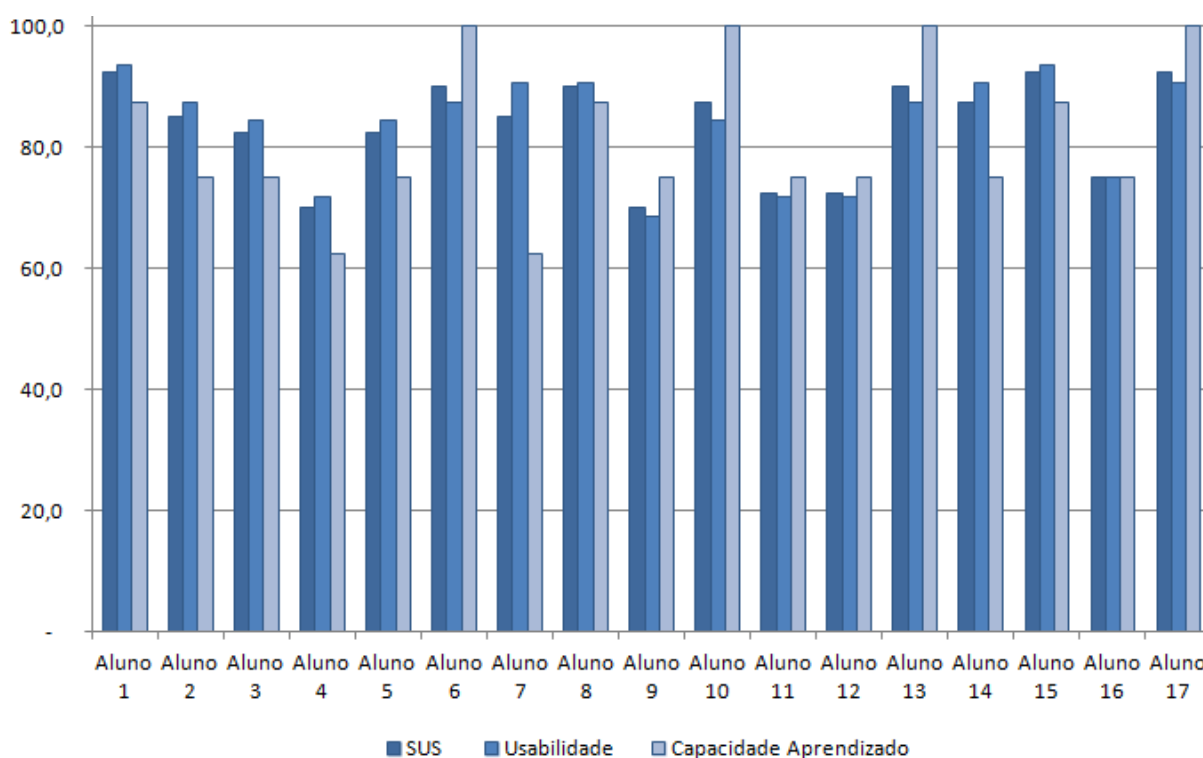
Tabela 13 – Classificação da usabilidade do aplicativo conforme duas escalas diferentes.

Sistema	Escore SUS	Classificação Bangor, Kortum e Miller		Classificação Sauro e Lewis	
		Faixa	Nota	Faixa	Nota
Aula de Campo	83,4	80 – 90	B (Excelente)	80,8 – 84	A

Fonte: Elaborada pelo próprio autor.

Conforme explicado anteriormente, a escala SUS contém questões específicas para permitir uma análise de forma separada das características de usabilidade e capacidade de aprendizado de um sistema. Considerando isso, a Figura 48 mostra um gráfico com um comparativo entre as médias individuais de todos os participantes do estudo com relação a essas características. Pode-se observar que, em geral, as três medidas têm valores aproximados para cada um dos alunos, ressaltando-se o aluno 16, que apresentou as três avaliações iguais (a 75,0). Também se percebe que as principais diferenças estão na dimensão Capacidade de Aprendizado, que representa os maiores (100,0) e menores (62,5) resultados do gráfico. Isso pode se justificar devido ao fato do cálculo dessa característica considerar apenas duas questões (4 e 10) entre as dez disponíveis na primeira parte do questionário. As respostas de todos os alunos às questões da primeira parte do instrumento de avaliação, incluindo os valores médios, encontram-se no APÊNDICE C – Respostas dos Alunos à Primeira Parte do Questionário de Avaliação desta tese de doutorado.

Figura 48 – Médias individuais dos alunos nas características avaliadas.



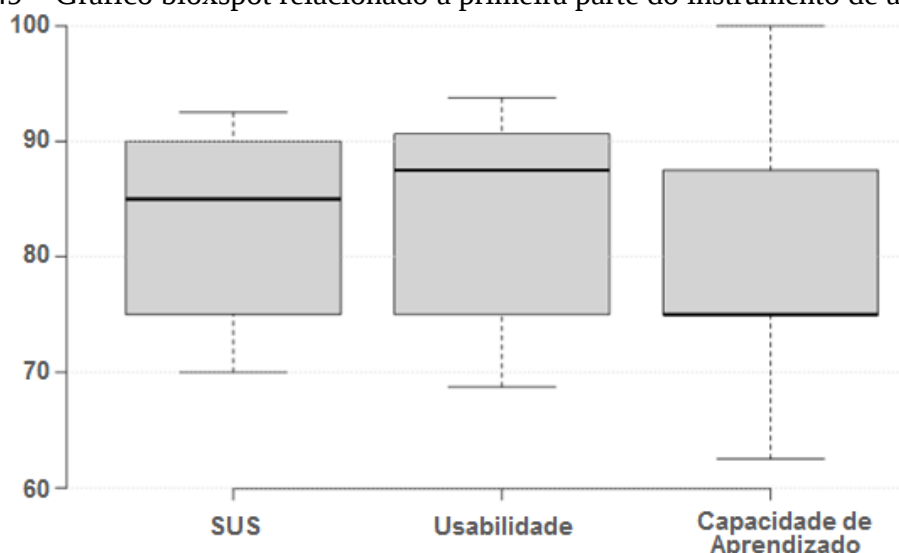
Fonte: Elaborada pelo próprio autor.

A Figura 49 apresenta um gráfico bloxspot que permite uma análise de forma integralizada da distribuição e simetria dos escores SUS, Usabilidade e Capacidade de Aprendizado. Esse gráfico reforça as constatações do gráfico anterior: uma distribuição

aproximada dos resultados das características (em torno de 80); e a Capacidade de Aprendizado é aquela que apresenta a maior diferença em termos de limite inferior e superior.

A partir da Tabela 14, pode-se observar que, tanto no resultado do escore médio SUS quanto da Usabilidade, os valores das medianas são superiores às médias. No caso da Capacidade de Aprendizado, isso se inverte. É importante destacar que nos gráficos boxsplot normalmente mostra-se a mediana (linha que secciona os retângulos), porém para avaliação dos resultados na escala SUS se usa a média. Neste estudo de caso específico, as três médias têm valores aproximados e acima de 80 (83,40, 83,9 e 81,6), reforçando o aspecto positivo da avaliação dessas características.

Figura 49 – Gráfico bloxspot relacionado à primeira parte do instrumento de avaliação.



Fonte: Elaborada pelo próprio autor.

Tabela 14 – Resumo dos dados referentes aos resultados da primeira parte da avaliação.

	SUS	Usabilidade	Capacidade de Aprendizado
Limite Superior	92,5	93,7	100,0
3º Quartil	90,0	90,6	87,5
Mediana	85,0	87,5	75,0
Média	83,4	83,8	81,6
1º Quartil	75,0	75,0	75,0
Limite Inferior	70,0	68,7	62,5

Fonte: Elaborada pelo próprio autor.

Segundo Davis (1989), a aceitação de uma aplicação está relacionada com a facilidade

de uso e a utilidade dela. Nesta tese de doutorado, a primeira parte do instrumento de avaliação focou na questão da usabilidade e a segunda parte era voltada para permitir a análise da utilidade da aplicação percebida pelos alunos. Diferentemente do SUS, as questões para se avaliar a utilidade percebida baseada no modelo de Davis podem variar e não existe uma fórmula padrão para se obter um resultado médio único de todas as questões. Assim, as pesquisas que utilizam o modelo de Davis realizam a avaliação dos questionários através da análise comparativa dos valores médios obtidos para cada questão e da frequência das respostas (ZBICK et al., 2015; MYERS; SCHROEDER, 2014; PARK et al., 2014; WENG; CHANG, 2014).

A Tabela 15 mostra os valores médios das respostas da segunda parte do questionário aplicado. Os bons resultados demonstram o aspecto da utilidade da aplicação percebida pelos alunos. O valor médio das questões redigidas positivamente (11, 13, 15 e 16) foi de 4,26. Dessas, apenas uma (Q16) obteve nota inferior a 4,0 (equivalente na escala Likert 5-pontos a opção “Concordo”), mesmo assim com o valor muito próximo a esse (3,94). Acredita-se que esse item obteve o menor resultado porque é o mais relacionado à questão da aprendizagem de conceitos teóricos, que é uma questão a ser melhor trabalhada com o professor especialista da área. Com relação às questões redigidas negativamente (12 e 14), os resultados demonstram que os alunos discordam que a aplicação dificulte o uso deles e complique a execução das atividades de aprendizagem (médias menores ou iguais a 2,0).

Tabela 15 – Resultado da avaliação da utilidade percebida da aplicação

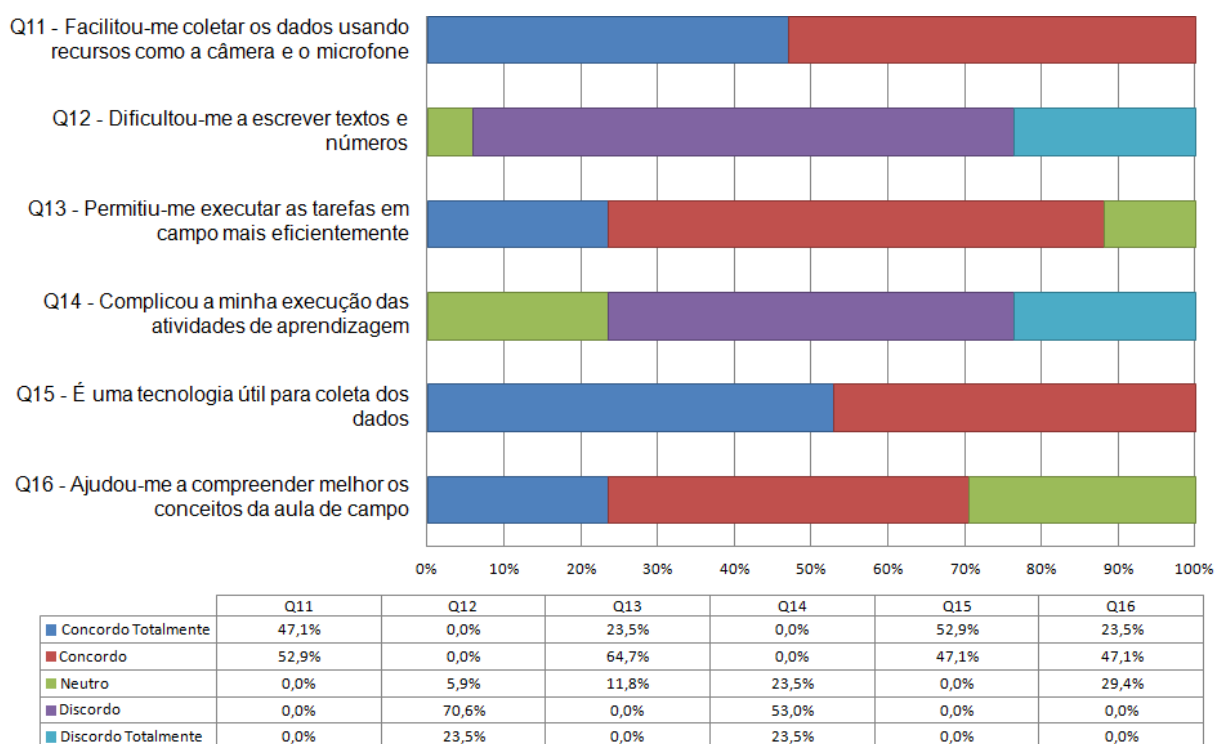
Questão	Média	Desvio Padrão
Q11 – Facilitou-me coletar os dados usando recursos como a câmera e o microfone.	4,47	0,51
Q12 – Dificultou-me a escrever textos e números.	1,82	0,53
Q13 – Permitiu-me executar as tarefas em campo mais eficientemente.	4,12	0,60
Q14 – Complicou a minha execução das atividades de aprendizagem.	2,00	0,71
Q15 – É uma tecnologia útil para coleta dos dados.	4,53	0,51
Q16 – Ajudou-me a compreender melhor os conceitos da aula de campo.	3,94	0,75

Fonte: Elaborada pelo próprio autor.

Ainda com relação à análise das respostas da segunda parte do questionário, a Figura

50 mostra a frequência (%) das respostas dos alunos referentes à utilidade da aplicação. Destacam-se positivamente as questões 11 e 15, onde 100% dos alunos responderam que concordam ou concordam plenamente que a aplicação é uma tecnologia útil para coleta de dados e facilita essa tarefa através do uso da câmera e do microfone do dispositivo móvel. Outro destaque está na questão 13, onde a maioria dos estudantes (88,2%) concorda que a aplicação permitiu que as tarefas em campo fossem executadas mais eficientemente (Q13).

Figura 50 – Frequência (%) das respostas da segunda parte do instrumento de avaliação.



Fonte: Elaborada pelo próprio autor.

Na terceira e última parte do questionário, os estudantes tinham a oportunidade de escrever observações sobre a aplicação. Na questão 17, deveriam responder sobre que forma (ou que formas) preferiu utilizar para responder as perguntas do aplicativo. A Tabela 16 mostra que a principal forma utilizada foi através da escrita de texto, ficando a captura de fotos da caderneta como segunda opção na preferência. A partir da leitura das respostas dessa questão, pôde-se concluir que o texto foi o preferido devido à praticidade e ao costume em usar essa forma de entrada no dispositivo móvel. Por outro lado, a gravação de áudio via microfone foi a opção menos usada em virtude de não ser um recurso usual para responder questões.

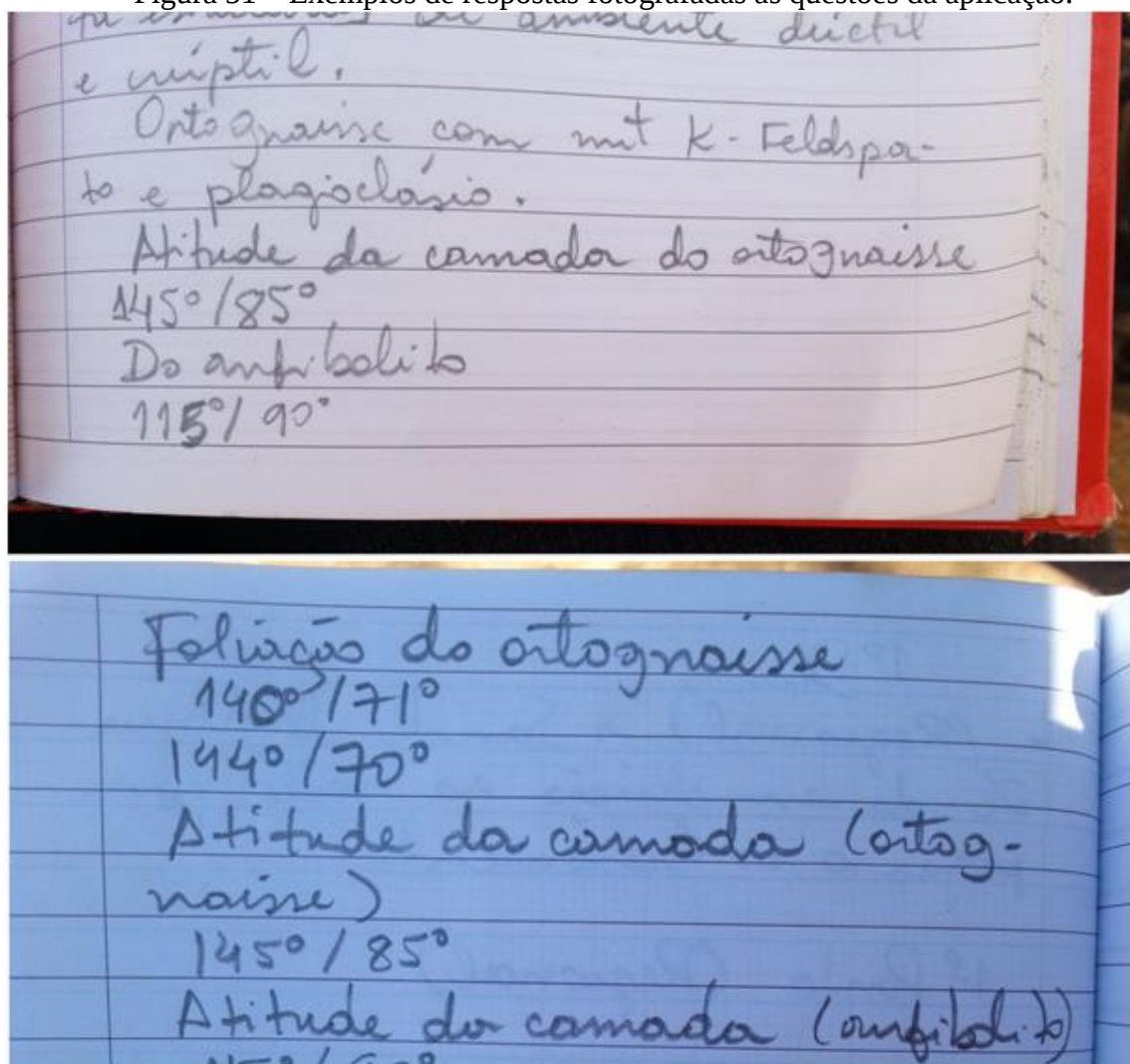
Tabela 16 – Forma escolhida para responder as perguntas

Opção Escolhida		
Texto	Foto da Caderneta	Gravação de Áudio
94,1%	41,2%	5,9%

Fonte: Elaborada pelo próprio autor.

A possibilidade de capturar fotos da caderneta de campo para responder as perguntas foi uma opção elogiada por alguns alunos, que justificaram o uso dessa opção à agilidade proporcionada por essa funcionalidade. A Figura 51 mostra exemplos de fotos das cadernetas capturadas pelos próprios alunos através da aplicação durante o estudo de caso. Essas imagens foram usadas como respostas às perguntas do aplicativo. Pode-se observar que as fotos permitem a leitura do conteúdo e reutilização do mesmo, seja para elaboração de relatórios de campo ou para avaliação pelo professor.

Figura 51 – Exemplos de respostas fotografadas às questões da aplicação.



Fonte: Elaborada pelo próprio autor.

A última questão do instrumento de avaliação com os alunos tratava-se de um espaço livre para eles escreverem sobre a aplicação e o uso dela em campo. Como pontos positivos, foram indicados: a facilidade para coletar os dados em campo; o fato de ser um meio a mais para armazenar em campo as informações da aula; que o conteúdo da aplicação ajudou a compreensão dos conceitos ensinados pelo professor durante a aula; as diferentes formas de armazenamento de informação (texto, áudio e foto); a possibilidade de gravar os dados geolocalizados (via GPS); e, a praticidade e agilidade para coletar as informações. Conforme respostas dos alunos, essas vantagens estão relacionadas ao fato do celular ser multifuncional e centralizar vários recursos (e.g. câmera, GPS, microfone) em um único dispositivo.

Com relação aos pontos negativos e às sugestões de melhoria, a principal reclamação dos alunos foi o fato da aplicação só permitir capturar uma foto por pergunta e uma foto por afloramento. Como os afloramentos são espaços extensos, os alunos sentiram falta de poder registrar várias fotos do local visitado. Entretanto, considerando o ambiente UFC-Inventor, esse é um problema fácil de resolver, pois basta ao professor adicionar novos recursos de captura de foto no modelo da aula de campo e gerar uma nova versão da aplicação.

Outros alunos também afirmaram que a aplicação poderia apresentar gráficos e tabelas úteis para a aula de campo. Eles afirmaram que como normalmente as regiões visitadas são as mesmas e já existe um grande volume de informações sobre elas, esses dados poderiam ser disponibilizados aos alunos durante as aulas de campo. Da mesma forma, esse problema também se resolve da mesma forma que o anterior, bastando apenas acrescentar o conteúdo adicional no modelo e gerar uma nova versão da aplicação. Outras sugestões feitas que também seriam resolvidas com alterações no modelo da aula de campo foram: mudança na ordem da apresentação dos textos, imagens e captura de fotos e exibição das coordenadas da localização.

Apenas uma reivindicação dos alunos, dentre as coletadas, não poderia ser resolvida pelo ambiente UFC-Inventor: mostrar no aplicativo um mapa com a localização do aluno. Como esse recurso não é suportado atualmente pela linguagem ML4UL, o ambiente não teria condições de gerar aplicações com ele. Entretanto, como a ML4UL trata-se de um modelo aberto, é possível especificar um novo elemento e, com pequenas alterações no UFC-Inventor, dar suporte à funcionalidade de mapa na aplicação.

6.4.4. Visualização dos Dados Transmitidos

Conforme Shakil, Faizi e Hafeez (2011), a importância de uma aula de campo não se

encerra com a visita em si, ocorrendo ainda uma fase pós-campo onde os dados coletados são analisados. Nesse momento, os alunos devem consolidar as informações e conhecimentos adquiridos durante a aula em campo e produzir relatórios que serão avaliados pelos professores das disciplinas.

Para possibilitar a visualização dos dados após a aula de campo, o UFC-Inventor se baseia principalmente em dois elementos da ML4UL: o <sensor>, que foi incluído no modelo da aula de campo e estava presente na aplicação para capturar a localização do estudante através do sensor GPS do dispositivo móvel; e o <synchronization>, que permitiu que os dados registrados pelos alunos fossem enviados para um servidor de banco de dados. É importante destacar que esse sistema de armazenamento de dados está previsto na especificação da ML4UL (Tabela 9), mas não faz parte do UFC-Inventor. Após a transmissão dos dados, o uso deles irá variar de acordo com as necessidades do usuário.

Nesta tese de doutorado, para demonstrar os benefícios da adoção do ambiente UFC-Inventor na fase pós-campo, optou-se por utilizar as *Google Fusion Tables*. Essa escolha ocorreu devido a dois motivos principais: por essa tecnologia já ser utilizada por geólogos em todo o mundo; e, por ela contar com o recurso para geração automática de mapa, a partir das coordenadas dos pontos armazenados no banco dados.

Foi desenvolvida uma página Web para acessar e exibir os dados que estavam na *Google Fusion Table* (ver

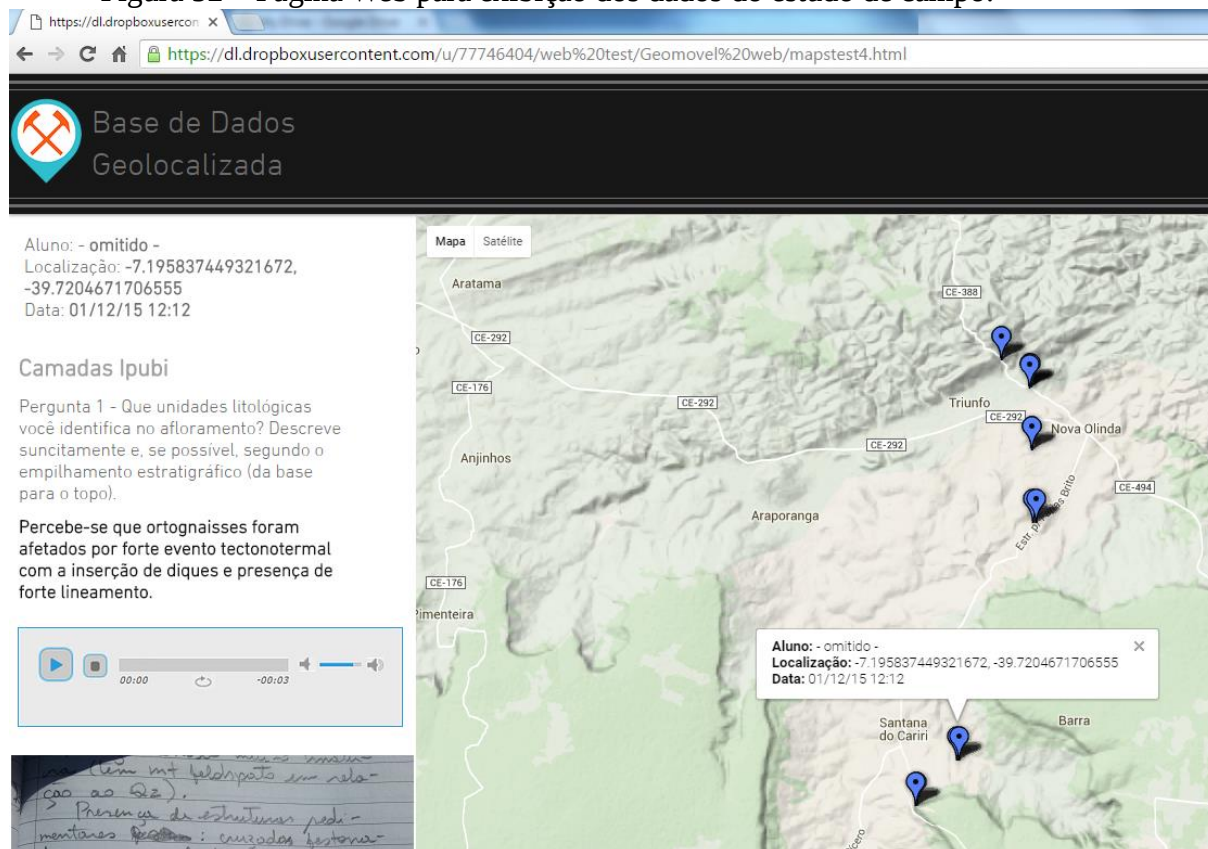
Figura 52). É possível observar os marcadores indicando os seis afloramentos que foram visitados. A figura também mostra um balão com as informações básicas sobre um dos pontos selecionados: o nome do aluno (omitido por questões de privacidade), a localização capturada daquele afloramento, e a data e hora do registro da informação no dispositivo móvel do estudante. As respostas do aluno para as perguntas sobre aquele afloramento selecionado (Camadas Ipubi), nas diferentes mídias (texto, áudio e foto), são apresentadas no lado esquerdo da página Web.

As Fusion Tables da Google foram utilizadas para armazenar os dados coletados. As informações que estavam armazenadas em um banco de dados genérico foram persistidas em uma Fusion Table, com estrutura específica para apresentação das informações daquela aula de campo.

Entretanto, a solução de sincronização dos dados proposta pelo ambiente UFC-Inventor, fundamentada na base de dados genérica, possibilita o uso de outras tecnologias, tais

como: a geração de arquivos no formato KMZ (para visualização no Google Earth) ou XLS (para visualização no Microsoft Excel); ou a consulta direta dos dados via site na Internet. Em ambos os casos, existe a necessidade do desenvolvimento de um sistema específico.

Figura 52 – Página Web para exibição dos dados do estudo de campo.



Fonte: Elaborada pelo próprio autor.

6.4.5. Visão do professor participante

Diferentemente da avaliação com os alunos, onde se aplicou um questionário de avaliação, no caso do professor foi realizada uma entrevista informal sobre o processo completo relacionado ao UFC-Inventor. Através dela, foi possível capturar a visão dele sobre o ambiente, que se deu em três aspectos principais: utilização da ferramenta de modelagem da aula de campo, observação do uso do aplicativo em campo pelos alunos e visualização dos dados coletados em campo.

Inicialmente, o professor de geologia afirmou que não teve muitas dificuldades em usar a ferramenta de autoria para modelar a aula de campo. O fato da interface do sistema ser baseada em um diagrama semelhante a fluxogramas tradicionais ajudou o processo de

modelagem. Ele também destacou a simplicidade para se adicionar fotos ao modelo da aula. Após visualizar a aplicação ubíqua gerada a partir do projeto da aula de campo, o professor ressaltou que a mesma refletia o modelo elaborado e atendia a expectativa. Entretanto, o professor afirma que o treinamento sobre o sistema recebido antes foi importante para conseguir modelar a aula e, para a adoção da ferramenta no departamento de Geologia, seria necessário um novo treinamento com os professores interessados.

Sobre o uso do aplicativo em campo pelos alunos, alguns aspectos chamaram a atenção do professor. Em primeiro lugar, nenhum aluno demonstrou ter encontrado dificuldades para instalar a aplicação. Se problemas ocorreram na instalação, os estudantes resolveram entre si, pois não recorreram nem a ele nem ao avaliador. Esse ponto foi importante e está relacionado à segunda observação dele: todos os alunos estavam usando a aplicação nas visitas aos afloramentos. Por serem alunos de semestres mais avançados do curso, imaginou-se que alguns não a usassem, uma vez que o uso do aplicativo não era obrigatório. A terceira e última observação do professor sobre o uso do aplicativo pelos alunos foi o fato deles a utilizarem para um fim não previsto inicialmente: como guia sobre a ordem de afloramentos já visitados e a visitar. Este fato é particularmente importante porque mostrou a ele que um aplicativo pode ter mais usos que o planejado, além de despertar para a inclusão dessa função (guia dos afloramentos) nas novas aulas de campo.

Após o retorno da aula de campo, o professor teve acesso aos dados coletados em campo e transmitidos pelos alunos. O uso da plataforma *Google Fusion Tables* foi elogiado pelo professor que informou já ter usado essa plataforma para armazenamento e visualização de dados. Outro ponto destacado pelo professor foi que, a partir do recurso de consulta dos dados coletados por todos os alunos através de um site na Internet, surgem diferentes possibilidades, como a elaboração de tabelas e gráficos individuais ou integralizados da turma e a avaliação das atividades realizadas em campo. Por fim, o professor concluiu informando que estava animado com os resultados do experimento e que espera utilizar o ambiente nas aulas de campo em outras disciplinas e em outras regiões geográficas com outros tipos de afloramentos.

6.4.6. Ameaças à validade do estudo de caso

Para a realização da avaliação com os alunos, diversos detalhes foram pensados de forma a tentar garantir a validade do estudo de caso. Foram seguidas orientações recomendadas por Sauro e Lewis (2012) que não comprometiam a amostra: a turma foi

composta com alunos com perfis similares e experientes em aulas de campo; o tamanho da amostra ($N = 17$) é uma quantidade aceita como válida para realização de cálculos estatísticos (SAURO, 2011); foi feita uma explanação sobre os objetivos do estudo e funcionamento do aplicativo, antes do início do teste; e, as tarefas solicitadas não eram complexas e repetiam-se, o que facilitava a utilização da aplicação. Por fim, para dar mais liberdade ao aluno, não era necessário que ele se identificasse no instrumento de avaliação.

Sobre o teste de campo, a presença do autor desta tese na aula de campo pode ter estimulado os alunos a usarem o aplicativo. A aula seguia essa rotina: os alunos, em grupos ou individualmente, caminhavam ao longo do afloramento escutando o professor de geologia e realizando anotações em suas cadernetas de campo. O aplicativo era usado em momentos diversos, às vezes no próprio local visitado e, em outros momentos, os alunos o utilizavam dentro do ônibus da universidade. Em todas essas situações, o autor desta tese mantinha-se distante, sem interagir com os alunos, apenas realizando suas observações. Portanto, acredita-se que ele não tenha interferido na utilização do aplicativo pelos alunos, tendo em vista que eles estavam distantes espalhados pelos afloramentos e todos os dezessete executaram as tarefas.

Outra preocupação diz respeito ao momento em que os alunos responderam o instrumento de avaliação do aplicativo. A aula começou por volta das 7h da manhã, os alunos visitaram os seis afloramentos, que ficavam quilômetros de distância entre si, em locais com temperaturas acima de 40° graus e só foram parar para almoçar por volta das 15:30. Então, o preenchimento dos questionários de avaliação aconteceu apenas depois das 16h, quando eles estavam cansados de um dia inteiro de aula de campo. Esse cansaço pode ter levado os alunos a responder o instrumento de avaliação de forma mais displicente. Entretanto, o esforço para concluir o questionário era pequeno, tendo em vista que das 18 questões 16 eram para marcar a opção correta. Além disso, o questionário tinha uma estrutura de perguntas escritas positivamente e negativamente que, caso o aluno respondesse sem atenção, estaria contradizendo-se. Inclusive, durante o preenchimento, alguns alunos erraram e pediram outros formulários para responder. E, como a análise dos questionários indicou uma uniformidade nas respostas, acredita-se que o cansaço não influenciou na resolução das questões. É importante destacar que os alunos não precisavam se identificar no formulário, dando uma maior liberdade a eles na hora de responder.

Com relação à avaliação com os professores, a escolha dos participantes do teste de usabilidade não ocorreu de forma aleatória. Três eram conhecidos do autor desta tese e os outros dois foram indicados por um dos três primeiros. De certa forma, isso pode ter

influenciado nos resultados dos testes. Contudo, foi perceptível que todos os professores conseguiram realizar de forma adequada as tarefas do experimento.

6.5. Conclusão

Este capítulo apresentou os detalhes sobre o processo de avaliação do UFC-Inventor, que englobou a análise de usabilidade do ambiente com professore e o estudo de caso com alunos em campo. Foram usados questionários padrões e cálculos estatísticos para se identificar os níveis proporcionados de usabilidade, utilidade e capacidade de aprendizado da aplicação gerada pelo UFC-Inventor, em uma situação de uso real.

Os resultados obtidos nos testes são animadores, devido principalmente aos seguintes fatores: os resultados favoráveis da avaliação da usabilidade do UFC-Inventor com os professores; as opiniões positivas obtidas através das respostas dos alunos nos instrumentos de avaliação; e a percepção otimista do professor participante do teste da aula de campo. No próximo capítulo são apresentadas as conclusões finais sobre esta tese de doutorado e os possíveis direcionamentos para trabalhos futuros.

7. CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo é dedicado às considerações finais desta tese de doutorado. Uma visão geral do trabalho é mostrada inicialmente na seção 7.1. A seção 7.2 apresenta os resultados alcançados e as principais contribuições alcançadas no decorrer do curso de doutorado deste candidato. Na seção 7.3 é feita uma análise da hipótese e das questões de pesquisa desta tese. Por fim, a seção 7.4 mostra possíveis trabalhos futuros decorrentes desta pesquisa de doutorado.

7.1. Visão geral do trabalho

Esta tese de doutorado envolveu um estudo bibliográfico aprofundado sobre os temas relacionados à pesquisa, o desenvolvimento e a avaliação de um ambiente computacional voltado para favorecer o uso das tecnologias da computação ubíqua em aulas de campo através da metodologia MDE. O primeiro capítulo desta tese introduziu o trabalho, mostrando a motivação, a hipótese e as questões de pesquisa.

Os dois capítulos seguintes foram voltados ao levantamento sobre os conceitos relacionados aos temas que embasam esta pesquisa (*Ubiquitous Learning* e *Model-Driven Engineering*) e outros trabalhos que apresentavam similaridades ao desenvolvido nesta tese. A partir do quarto capítulo iniciaram-se as contribuições dessa pesquisa. Nesse capítulo, foi apresentado um mapeamento sistemático com experiências de aulas de campo suportadas por recursos de computação ubíqua e um conjunto de requisitos fundamentais para esse tipo de prática educativa.

No quinto capítulo, foram apresentadas as principais contribuições desse trabalho, que consistem no ambiente proposto nesta tese (UFC-Inventor) e em seus componentes (linguagem ML4UL e os módulos UFC-GLM e UFC-Generator). A avaliação do funcionamento completo do UFC-Inventor foi apresentada no sexto capítulo. O sétimo capítulo conclui esse trabalho apresentando as principais contribuições, análise da hipótese e das questões de pesquisa e os trabalhos futuros.

7.2. Principais contribuições e resultados alcançados

As principais contribuições desta tese de doutorado estão diretamente relacionadas com as metas estabelecidas no capítulo de introdução e são descritas a seguir:

1. **O conjunto de requisitos de aulas de campo ubíquas**, que são fundamentais em sistemas voltados para esse tipo de prática educativa e consistiram na base para construção do ambiente UFC-Inventor. Podem ser utilizados como referência para o desenvolvimento de aplicações ubíquas voltadas para aula de campo.
2. **A Linguagem Específica de Domínio ML4UL**, que consiste em um metamodelo aberto e independente de estratégia pedagógica, cuja principal vantagem é poder ser utilizado para modelar aulas de campo suportadas por recursos da computação ubíqua. Essa DSL pode ser reutilizada em outras pesquisas para servir como modelo de representação de aplicações ubíquas em diferentes contextos.
3. **A Ferramenta de Modelagem UFC-GLM**, que permite que profissionais de ensino, com pouco ou nenhum conhecimento de programação, construam projetos gráficos de aulas de campo que incluam recursos de computação ubíqua. Entre algumas das aplicações possíveis da UFC-GLM, posteriormente, ela pode ser adotada em pesquisas para geração de projetos ML4UL similares ao proposto nesta tese.
4. **O Sistema de Geração de Aplicações Multiplataforma Ubíquas UFC-Generator**, que gera aplicações ubíquas para diferentes sistemas operacionais, conforme características estabelecidas no modelo da aula de campo. Além do uso descrito nesta tese e considerando o fato da ML4UL ser uma especificação aberta, o UFC-Generator pode ser utilizado para gerar aplicações multiplataforma a partir de modelos elaborados a partir de outras ferramentas, que não a UFC-GLM.
5. **O Ambiente Integrado de Modelagem e Geração de Aplicações UFC-Inventor**, que, baseado em uma linguagem específica de domínio, possibilita de forma integrada a modelagem gráfica de aulas de campo com recursos da computação ubíqua e a geração de aplicações para diferentes sistemas operacionais móveis.

Em comparação com trabalhos apresentados na Tabela 3, a Tabela 17 replica esse quadro adicionando uma coluna para o ambiente UFC-Inventor. Das informações apresentadas na tabela, destaca-se o fato do UFC-Inventor ser o único ambiente que permite modelagem gráfica através de uma especificação aberta que possibilita a interoperabilidade com outros sistemas, através da linguagem ML4UL. Além disso, é importante ressaltar que as aplicações geradas pelo UFC-Inventor não estão restritas a um único Sistema Operacional, diferentemente dos outros trabalhos analisados. Esta foi uma qualidade importante durante o estudo de caso, porque permitiu a todos os alunos usarem a aplicação gerada de forma individualizada. Com relação à sensibilidade ao contexto, o UFC-Inventor oferece suporte a

diferentes tipos de sensores e de informações contextuais, podendo ainda ser estendido para incorporar outros tipos. Por fim, ressalta-se o fato do UFC-Inventor possibilitar que as aplicações geradas funcionem sem a necessidade de conexão com a Internet. Esta característica foi fundamental para o sucesso do estudo de caso, pois em vários momentos da aula de campo não havia conexão de dados com a Internet.

Tabela 17 – Comparação entre os trabalhos relacionados e o UFC-Inventor.

Ferramentas	LEMONADE	UoLmP	MAT for ARLearn	Midgar	App Inventor	UFC-Inventor
Modelagem gráfica	Sim	Não	Sim	Sim	Sim	Sim
Conhecimentos de programação	Não	Sim	Não	Não	Sim	Não
Utilização de Especificações	Não	IMS LD	Não	Não	Não	IMS LD / ML4UL
Modelo aberto	Não	Sim	Não	Não	Não	Sim
Compartilhamento	Não	Não	Sim	Sim	Sim	Sim
Sistema operacional móvel	Windows	Android	Android	Android	Android	Android, Windows Phone, iOS, entre outros.
Uso de sensores	GPS	*	GPS	**	GPS, NFC, Bússola e Acelerômetro	GPS, QR Code, Bússola, ***
Informação Contextual	Localização	*	Localização	**	Localização e Orientação	Localização, Código QR, Orientação, ***
Dependência de conexão	Não	Sim	Sim	Sim	Não	Não
Interação sem fio	Não	Não	Não	Não	Sim	Sim

Fonte: elaborada pelo próprio autor.

* Como a abordagem proposta no trabalho UoLmP (GÓMEZ, 2013) não prevê uma ferramenta de edição, não é possível a inclusão direta no projeto de sensores. Entretanto, a solução teórica apresentada prevê um modelo contextual baseado nas informações de localização, perfil do usuário, tempo, dispositivo móvel e informações ambientais (e.g. temperatura e umidade).

** Os tipos de sensores e as informações contextuais vão depender dos Smart Objects disponibilizados no servidor do Midgar (GARCÍA et al., 2014).

*** A linguagem ML4UL e o ambiente UFC-Inventor podem ser estendidos para suportar outros sensores e informações contextuais. Além disso, através das perguntas, que são suportadas tanto pela linguagem quanto pelo ambiente, é possível criar sensores virtuais que podem ser usados na aplicação para obter informações contextuais como o sexo e a idade do usuário.

Com relação aos artigos científicos, a Tabela 18 apresenta os trabalhos produzidos até o momento, sendo seis aceitos e dois elaborados. Todos são trabalhos completos (*full paper*), têm relação com as pesquisas desenvolvidas pelo autor nesta tese de doutorado e o mesmo está como primeiro autor.

Tabela 18 – Artigos científicos produzidos

#	Dados do Artigo	Qualis	Situação
1	2013 Simpósio de Doutorado (WTDSOFT) do CBSOFT Uma Abordagem Dirigida por Modelos para a Geração de Sistemas Ubíquos de Aprendizagem.	-	Publicado
2	2013 Simpósio Brasileiro de Informática na Educação – SBIE Geomóvel: Um Aplicativo para Auxílio a Aulas de Campo de Geologia	B2	Publicado
3	2014 IEEE Frontiers in Education (FIE) Conference A Mobile learning System to Enhance Field Trips in Geology	B1	Publicado
4	2015 Simpósio Brasileiro de Informática na Educação – SBIE Aulas de Campo Ubíquas	B2	Publicado
5	2015 Simpósio Brasileiro de Informática na Educação – SBIE ML4UL: Uma Extensão ao Padrão IMS LD para Modelagem de Sistemas Ubíquos para Aulas de Campo	B2	Publicado
6	2016 ACM Symposium On Applied Computing – SAC Mobile Phone Text Messaging to Increase Student Participation: An Experience in a Blended Course	A1	Aceito
7	2016 Elsevier Computers in Human Behavior UFC-Inventor: An environment for teachers design and deploy ubiquitous applications for field trips	-	Elaborado
8	2016 Concurso de Teses e Dissertações (CTD) do Simpósio Brasileiro de Informática na Educação – SBIE UFC-Inventor: um ambiente para modelagem e geração de aplicações para aulas de campo ubíquas	-	Elaborado

Fonte: elaborada pelo próprio autor.

O primeiro artigo mostra as ideias gerais da tese de doutorado e foi apresentado no Workshop de Teses e Dissertações do Congresso Brasileiro de Software em 2013. No segundo, uma aplicação ubíqua para aulas de campo de geologia desenvolvida nessa pesquisa

de doutorado é apresentada: o Geomóvel. O terceiro artigo mostra os detalhes e os resultados da utilização do Geomóvel em aulas de um curso de graduação em geologia da UFC. O quarto e o quinto artigo estão diretamente relacionados ao doutorado, correspondendo respectivamente ao Capítulo 4 à seção 5.2. O sexto artigo apresenta um trabalho parcialmente relacionado com essa pesquisa de doutorado, com foco em *Mobile Learning* através das mensagens de celular.

O sétimo e o oitavo artigos refletem os resultados completos dessa pesquisa de doutorado, sendo um internacional e outro nacional. Até o momento da conclusão do texto desta tese, ambos ainda não haviam sido submetidos, encontrando-se em versões iniciais.

7.3. Análise da hipótese e das questões de pesquisa

A seguir são apresentadas as respostas a cada uma das questões de pesquisa que nortearam o desenvolvimento desta tese de doutorado.

QP01: Quais são os principais requisitos que uma aplicação ubíqua deve ter para auxiliar aulas de campo?

Resposta: A partir do mapeamento sistemático apresentado no Capítulo 4, foi possível obter uma visão geral do perfil das aplicações ubíquas usadas em aula de campo, e, então, identificar um conjunto de requisitos essenciais para esse tipo de aplicação, que é apresentado na Tabela 6.

QP02: Como especificar as aulas de campo ubíquas de modo que seja possível a representação de atividades de aprendizagem de maneira estruturada, aberta, semântica e legível via computador?

Resposta: Através de modelos usando a linguagem ML4UL, que estende a especificação IMS LD, é possível representar as atividades de aprendizagem de aulas de campo de forma estruturada, aberta, semântica e legível por computador, através da codificação em arquivos XML. Além disso, por serem baseadas na IMS LD, aulas especificadas em ML4UL poderiam adotar diferentes abordagens pedagógicas (e.g. behaviorismo, cognitivismo, construtivismo). Porém, o suporte à diversidade pedagógica não foi testado nesse trabalho de doutorado.

QP03: É possível desenvolver uma ferramenta de modelagem que possibilite a representação gráfica de aulas de campo ubíquas por profissionais de ensino com pouco ou nenhum conhecimento sobre linguagens de programação?

Resposta: A avaliação de usabilidade do UFC-Inventor e o estudo de caso mostraram que professores, sem conhecimentos de programação, conseguiram usar o ambiente, modelar aulas de campo com recursos de computação ubíqua e gerar aplicações para dispositivos móveis. As respostas aos questionários aplicados com eles indicam uma boa aceitação do ambiente

QP04: Como possibilitar a geração automática de aplicações fáceis de usar, úteis e compatíveis com diferentes sistemas operacionais, a partir das especificações das aulas de campo ubíquas?

Resposta: Utilizando os modelos da ML4UL para especificar as aulas de campo e, posteriormente, executando o módulo UFC-Generator para gerar a aplicação ubíqua no sistema operacional alvo. O estudo de caso realizado demonstrou a usabilidade, simplicidade de aprendizado e utilidade de uma aplicação gerada pelo UFC-Generator, a partir de um modelo produzido pela ferramenta de modelagem gráfica UFC-GLM.

A confirmação da hipótese investigada nesta tese é analisada a partir dos resultados obtidos ao longo desse doutorado e das constatações observadas a partir da avaliação da usabilidade feita com os professores e do estudo de caso realizado em aula de campo real.

Hipótese: Um ambiente integrado que possibilita o projeto gráfico de aulas de campo suportadas por recursos de computação ubíqua é capaz de produzir modelos abertos e, a partir deles, gerar aplicações fáceis de usar e úteis para os alunos em campo utilizando dispositivos móveis com diferentes sistemas operacionais.

Resultado: Confirmada. Inicialmente, conforme descrito no Capítulo 6, professores modelaram graficamente aulas de campo e atestaram, a partir de questionários, a usabilidade do ambiente. Além disso, um professor de geologia projetou graficamente uma aula de campo com recursos de computação ubíqua, gerando um modelo na linguagem ML4UL. Através desse modelo, foi possível gerar códigos-fonte e criar aplicações para dispositivos móveis de dois sistemas operacionais diferentes. Por fim, em uma aula de campo real, alunos utilizaram seus dispositivos móveis com a aplicação gerada e constataram a facilidade de uso, a simplicidade para aprendizado e a utilidade em campo da mesma.

7.4. Limitações da Tese

Ao longo desse trabalho de Doutorado, surgiram fatores que foram definindo e limitando o escopo desta tese, seja por dificuldades técnicas ou escolhas de implementação. Inicialmente, é importante destacar que o ambiente foi proposto considerando requisitos genéricos para aulas de campo de diferentes áreas, porém a avaliação foi realizada particularmente com uma turma de alunos do curso de geologia. Desta forma, mesmo tendo sido realizado testes de usabilidade com professores de outras áreas (zootecnia, geografia e biologia), não podemos afirmar que os mesmos resultados devem ser alcançados com alunos de outros domínios de conhecimento. Também vale mencionar que a escolha dos professores para o teste do UFC Inventor se deu por proximidade com o autor desta tese, o que pode implicar em uma ameaça a essa validação específica.

Esta tese de doutorado tem um caráter computacional e baseia-se em uma hipótese onde o objetivo principal é comprovar a eficiência do ambiente sob o aspecto do funcionamento da tecnologia e da abordagem proposta. Desta forma, ela não inclui a verificação dos aspectos relacionados à aprendizagem proporcionados pelo uso das aplicações geradas. A avaliação realizada envolveu os aspectos de usabilidade, facilidade de aprendizado e utilidade.

Com relação à ferramenta de modelagem são realizadas verificações simples relativas à construção do modelo, como a identificação de atividades sem conexão. Entretanto, a UFC-GLM faz uma análise da corretude das regras especificadas nas atividades sensíveis ao contexto (elemento *<contextaware-activity>*). Assim, por exemplo, duas atividades podem conter especificações com regras conflituosas, que podem gerar comportamentos não esperados na aplicação gerada.

O UFC-Inventor foi desenvolvido de forma a viabilizar a geração de aplicações que independem da Internet para funcionar, devido à possibilidade do uso delas em locais distantes e sem conexão. Por outro lado, não foram previstos recursos para integração com soluções do proeminente paradigma Internet das Coisas, que poderiam ampliar as oportunidades de utilização do UFC-Inventor, como em espaços inteligentes.

Como descrito no capítulo anterior, não é possível realizar a consulta (pós-campo) das informações que foram registradas durante a aula de campo através dos recursos do UFC-Inventor. O ambiente não prevê um processo automatizado para visualização dos dados após

o retorno do campo. O escopo do UFC-Inventor vai até a sincronização das informações coletadas em uma base de dados remota. A forma como essas informações serão disponibilizadas posteriormente irá depender de implementações específicas.

A ML4UL é uma linguagem específica de domínio aberta que pode ser estendida para incorporar novos recursos. A UFC-GLM é uma ferramenta de modelagem baseada na ML4UL, porém com implementação restrita a alguns elementos da linguagem (descritos no Capítulo 5). Assim, acréscimos de novos elementos na ML4UL implicam na necessidade de alterações e uma nova versão da UFC-GLM.

O UFC-Inventor possibilita a geração de aplicações com versões diferentes que variam o estilo visual. Entretanto, nesta tese de doutorado não houve um estudo aprofundado sobre a questão dos layouts das aplicações móveis. Os estilos disponíveis no UFC-Generator consistem em modelos simples para demonstração da hipótese proposta, sem se preocupar com recomendações estéticas.

A ferramenta UFC-GLM consiste em um software para a plataforma Windows® de uso individual. O projetista pode criar modelos de aula de campo e enviar os arquivos para outros profissionais de forma a permitir o reuso dos mesmos. Entretanto, não é possível que vários usuários trabalhem de forma colaborativa e editem o modelo ao mesmo tempo em um espaço centralizado na Internet. Além disso, outro obstáculo para ampliação do uso do UFC-Inventor é a dificuldade de instalação do ambiente, que requer o envolvimento de usuários especialistas em todas as ferramentas e componentes utilizados. Isso pode dificultar a replicabilidade da solução.

7.5. Trabalhos futuros

Considerando-se as limitações citadas na seção anterior e oportunidades relacionadas de pesquisas interessantes, a seguir são listados trabalhos futuros que podem melhorar e evoluir as contribuições desta tese:

- **Promover uma utilização maior e analisar os ganhos de aprendizagem das aplicações geradas.** Uma questão mais relacionada à área educacional diz respeito ao ganho de aprendizagem proporcionado pelo uso de uma aplicação gerada pelo UFC-Inventor. Este é um aspecto complexo e está associado ao desempenho do professor no uso do ambiente. Entretanto, trata-se de uma pesquisa interessante no momento em que os resultados positivos podem promover o uso das tecnologias ubíquas em aulas

de campo. Para isso, são necessários mais experimentos com aplicações distintas, inclusive em domínios de conhecimento diferentes.

- **Analisar a corretude dos modelos elaborados.** No momento em que se considera um uso maior do ambiente, surge a necessidade de se verificar se os modelos que estão sendo elaborados estão consistentes. Esta não é uma tarefa simples porque envolve a verificação tanto de aspectos pedagógicos, como os objetivos de aprendizagem da aula de campo projetada, quanto de questões tecnológica, como a validade das regras contextuais especificadas.
- **Integrar o ambiente UFC-Inventor com as tecnologias da Internet das Coisas.** Uma tecnologia que tem ganhado a atenção dos pesquisadores nos últimos anos e tem muitos pontos em comum com a computação ubíqua é o paradigma Internet das Coisas. Sendo assim, a integração do UFC-Inventor com as soluções desse paradigma mostra-se como uma alternativa de pesquisa interessante ao possibilitar o uso do ambiente em cenários não previstos inicialmente.
- **Automatizar a visualização dos dados transmitidos.** Como demonstrado no estudo de caso desta tese, foi necessário o desenvolvimento de um sistema específico para possibilitar o acesso aos dados registrados durante a aula de campo. Portanto, apresenta-se como uma possibilidade de pesquisa futura interessante a implementação de uma solução para permitir que os dados registrados em campo pelas aplicações possam ser visualizados. Esta não é uma tarefa trivial principalmente porque duas informações essenciais variam de projeto para projeto: o modelo de dados da aplicação de coleta e o formato de saída dos dados esperado pelo professor.
- **Aperfeiçoar o processo de incorporação de novos recursos de computação ubíqua.** Atualmente, para se incorporar novos elementos da ML4UL na UFC-GLM é necessário alterar o código-fonte da ferramenta e gerar uma nova versão. Portanto, para tornar a ferramenta de modelagem e, conseqüentemente, o ambiente UFC-Inventor, mais flexível, uma possibilidade seria desenvolver uma forma para aperfeiçoar a adição de novos recursos ao ambiente sem ter que mudar o seu código-fonte. Para isso, uma solução seria a utilização de documentos *templates* que já teriam uma estrutura padrão que fosse reconhecida automaticamente pela ferramenta, através da utilização de atributos como: nome do recurso, propriedades, valores aceitáveis, entre outros. Porém, para criação desses *templates*, o usuário necessita ter conhecimentos técnicos sobre o funcionamento do recurso que está sendo adicionado.
- **Desenvolver e experimentar novos layouts de interface.** Nesta tese de Doutorado foram implementados apenas dois tipos diferentes de layouts de interface, que

variavam pouco e tinham como objetivo principal demonstrar o funcionamento da proposta. Portanto, essa é uma área que apresenta várias possibilidades de investigação, como a geração de layouts mais intuitivos e que facilitem o uso.

- **Migrar a ferramenta de modelagem para uma plataforma colaborativa e na Web.** Essa é uma oportunidade mais focada em implementação, que está relacionada à dificuldade de uso de um software desktop por várias pessoas dispersas geograficamente, com a possibilidade de visualizar a evolução da interface. Ao mesmo tempo em que produziria benefícios importantes, como evitar a necessidade de instalação, essa possibilidade de trabalho futuro impõe questões que devem ser consideradas, como controle de acesso, permissões dos usuários/grupos, versionamento do projeto de aula, entre outras.

REFERÊNCIAS

- ABOWD, Gregory D.; MYNATT, Elizabeth D. Charting past, present, and future research in ubiquitous computing. **ACM Transactions on Computer-Human Interaction (TOCHI)**, v. 7, n. 1, p. 29-58, 2000.
- AINSWORTH, Shaaron; GRIMSHAW, Shirley. Evaluating the REDEEM authoring tool: can teachers create effective learning environments?. **IJ Artificial Intelligence in Education**, v. 14, n. 3-4, p. 279-312, 2004.
- AKMAN, Ibrahim; TURHAN, Cigdem. User acceptance of social learning systems in higher education: an application of the extended Technology Acceptance Model. **Innovations in Education and Teaching International**, p. 1-9, 2015.
- ALAMER, Reem A.; AL-OTAIBI, Hind M.; AL-KHALIFA, Hend S. L3MS: A Lightweight Language Learning Management System Using Mobile Web Technologies. In: **Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on**. IEEE, 2015. p. 326-327.
- ALLWIHAN, Regad; BRAILSFORD, Tim; COBB, Sue. Capturing experience in the field trip: a comparison study of using mobile devices between geography and architecture students. **INTED2013 Proceedings**, p. 1550-1556, 2013.
- ATOJI, Rodolpho Iemini. Bluetooth e NFC: estudo de caso. **Universidade de São Paulo Instituto de Matemática e Estatística, São Paulo**, 2010.
- ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.
- BALASUBRAMANIAN, Krishnakumar *et al.* Component-based system integration via (meta) model composition. In: **Engineering of Computer-Based Systems, 2007. ECBS'07. 14th Annual IEEE International Conference and Workshops on the**. IEEE, 2007. p. 93-102.
- BANGOR, Aaron; KORTUM, Philip; MILLER, James. Determining what individual SUS scores mean: Adding an adjective rating scale. **Journal of usability studies**, v. 4, n. 3, p. 114-123, 2009.
- BARAKI, Harun *et al.* **Towards Interdisciplinary Design Patterns for Ubiquitous Computing Applications**. kassel university press GmbH, 2014.
- BARKER, Phil. What is IEEE learning object metadata/IMS learning resource metadata?. **cetis standards briefings series**, 2005.

BATTISTELLA, Paulo Eduardo; VON WANGENHEIM, Aldo. Avaliação de Ferramentas de Autoria Gratuitas para produção de Objetos de Aprendizagem no padrão SCORM. **Revista Brasileira de Informática na Educação**, v. 19, n. 3, p. 16-28, 2011.

BEHRENDT, Marc; FRANKLIN, Teresa. A Review of Research on School Field Trips and Their Value in Education. **International Journal of Environmental and Science Education**, v. 9, n. 3, p. 235-245, 2014.

BELLAVISTA, Paolo *et al.* A survey of context data distribution for mobile ubiquitous systems. **ACM Computing Surveys (CSUR)**, v. 44, n. 4, p. 24, 2012.

BERTRAN, Benjamin *et al.* DiaSuite: A tool suite to develop Sense/Compute/Control applications. **Science of Computer Programming**, v. 79, p. 39-51, 2014.

BEZERRA, Carla *et al.* Challenges for usability testing in ubiquitous systems. In: **Proceedings of the 26th Conference on l'Interaction Homme-Machine**. ACM, 2014. p. 183-188.

BONETT, Douglas G.; WRIGHT, Thomas A. Cronbach's alpha reliability: Interval estimation, hypothesis testing, and sample size planning. **Journal of Organizational Behavior**, v. 36, n. 1, p. 3-15, 2015.

BOYINBODE, O. K.; AKINTOLA, K. G. Effecting E-Learning with *U-learning* Technology in Nigerian Educational System. **E-LEARNING**, v. 1, p. 1, 2009.

BOYLE, Robin; INGHAM, Joanne M. Suggestions on How to Conduct Empirical Research: A Behind-the-Scenes View. **Perspectives: Teaching Legal Research and Writing**, v. 15, n. 3, p. 176, 2007.

BRAMBILLA, Marco; CABOT, Jordi; WIMMER, Manuel. Model-driven software engineering in practice. **Synthesis Lectures on Software Engineering**, v. 1, n. 1, p. 1-182, 2012.

BRITAIN, Sandy. A review of learning design: concept, specifications and tools. **A report for the JISC E-learning Pedagogy Programme**, v. 2006, 2004.

BROOKE, John. SUS-A quick and dirty usability scale. **Usability evaluation in industry**, v. 189, n. 194, p. 4-7, 1996.

BURGOS, Daniel; TATTERSALL, Colin; KOPER, Rob. Re-purposing existing generic games and simulations for e-learning. **Computers in Human Behavior**, v. 23, n. 6, p. 2656-2667, 2007.

BURGOS, Daniel. What is wrong with the IMS Learning Design specification? Constraints

And Rec-ommendations. **LWA 2010**, p. 281, 2010.

BURGOS, Daniel. How to use IMS LD to support eLearning in an effective way (invited talk). In: **Computers in Education (SIIE), 2012 International Symposium on**. IEEE, 2012. p. 1-1.

BURSZTYN, Natalie et al. Utilizing geo-referenced mobile game technology for universally accessible virtual geology field trips. **International Journal of Education in Mathematics, Science and Technology**, v. 3, n. 2, p. 93-100, 2015.

CARVALHO, Windson et al. Towards context-aware and mobile e-learning applications. 2011. In: TISE 2011, **Congreso Internacional de Informática Educativa**, 2011, Santiago, Chile.

CASSOU, Damien et al. A generative programming approach to developing pervasive computing systems. In: **ACM Sigplan Notices**. ACM, 2009. p. 137-146.

CASSOU, Damien *et al.* Toward a tool-based development methodology for pervasive computing applications. **Software Engineering, IEEE Transactions on**, v. 38, n. 6, p. 1445-1463, 2012.

CHANG, En-Chi; HUANG, Chia-Yin. Technology Acceptance Model, Consumer Personality and Smartphone Users' Satisfaction. In: **Marketing Dynamism & Sustainability: Things Change, Things Stay the Same....** Springer International Publishing, 2015. p. 710-712.

CHEN, Baiyun; DENOYELLES, Aimee. Exploring students' *mobile learning* practices in higher education. **Educause Review**. Disponível em <http://www.educause.edu/ero/article/exploring-students-mobile-learning-practices-higher-education>, 2013.

CHEN, Chia-Chen; HUANG, Tien-Chi. Learning in a u-Museum: Developing a context-aware *ubiquitous learning* environment. **Computers & Education**, v. 59, n. 3, p. 873-883, 2012.

CHEN, Chih-Hung; LIU, Guan-Zhi; HWANG, Gwo-Jen. Interaction between gaming and multistage guiding strategies on students' field trip mobile learning performance and motivation. **British Journal of Educational Technology**, 2015.

CHIANG, Tosti HC; YANG, Stephen JH; HWANG, Gwo-Jen. Students' online interactive patterns in augmented reality-based inquiry activities. **Computers & Education**, v. 78, p. 97-108, 2014.

CHIN, Kai-Yi; CHEN, Yen-Lin. A *mobile learning* support system for *ubiquitous learning* environments. **Procedia-Social and Behavioral Sciences**, v. 73, p. 14-21, 2013.

CHUNG, Hsin-Hui; CHEN, Shu-Chu; KUO, Min-Hsiu. A Study of EFL College Students' Acceptance of Mobile Learning. **Procedia-Social and Behavioral Sciences**, v. 176, p. 333-339, 2015.

COHEN, Eli; NYCZ, Malgorzata. Learning objects and e-learning: An informing science perspective. **Interdisciplinary Journal of E-Learning and Learning Objects**, v. 2, n. 1, p. 23-34, 2006.

COHEN, Philip R. et al. Sketch-Thru-Plan: a multimodal interface for command and control. *Communications of the ACM*, v. 58, n. 4, p. 56-65, 2015.

CONOLE, Gráinne *et al.* Visualising learning design to foster and support good practice and creativity. **Educational Media International**, v. 45, n. 3, p. 177-194, 2008.

COPE, Bill; KALANTZIS, Mary. *Ubiquitous learning: An agenda for educational transformation*. **Proceedings of the 6th Networked Learning, Greece**, 2008.

CUADRADO, Jesus Sanchez; IZQUIERDO, Javier Luis Cánovas; MOLINA, Jesus Garcia. Applying model-driven engineering in small software enterprises. **Science of Computer Programming**, v. 89, p. 176-198, 2014.

DAVIES, Sarah-Jane *et al.* Enabling remote activity: using mobile technology for remote participation in geoscience fieldwork. **Proceedings of the European Geosciences Union General Assembly**, 2010.

DAVIS, Fred D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. **MIS quarterly**, p. 319-340, 1989.

DE-LA-FUENTE-VALENTÍN, Luis; PARDO, Abelardo; KLOOS, Carlos Delgado. Generic service integration in adaptive learning experiences using IMS learning design. **Computers & Education**, v. 57, n. 1, p. 1160-1170, 2011.

DEN HAAN, Johann. MDE-Model Driven Engineering-reference guide. URL: <http://www.theenterpriseearchitect.eu/blog/2009/01/15/mde-model-driven-engineering-reference-guide/>, 2009. Último acesso em 15 de maio de 2015.

DERNTL, Michael *et al.* The conceptual structure of IMS Learning Design does not impede its use for authoring. **Learning Technologies, IEEE Transactions on**, v. 5, n. 1, p. 74-86, 2012.

DIRENE, Alexandre *et al.* Acquiring expertise in medical radiology through long-term interactions. In: **Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on**. IEEE, 2008. p. 403-408.

DLUDLA, Angeline G. *et al.* System architecture for ubiquitous live video streaming in university network environment. In: **AFRICON, 2013**. IEEE, 2013. p. 1-5.

DONOHOO, Brad K.; OHLSEN, Chris; PASRICHA, Sudeep. AURA: An application and user interaction aware middleware framework for energy optimization in mobile devices. In: **Computer Design (ICCD), 2011 IEEE 29th International Conference on**. IEEE, 2011. p. 168-174.

FAVRE, Jean-Marie. Foundations of model (driven)(reverse) engineering: Models. In: **Proceedings of the International Seminar on Language Engineering for Model-Driven Software Development, Dagstuhl Seminar 04101**. 2004.

FIGUEIREDO, Vânia Santos; SILVA, Geane Suelí Castro. A importância da aula de campo na prática em geografia. **10º Encontro Nacional de Prática de Ensino em Geografia**. Campo Grande: UFCG, 2009.

FLÔRES, Maria Lucia Pozzatti; TAROUCO, Liane Margarida Rockenbach; REATEGUI, Eliseo Berni. Funcionalidades da ferramenta de autoria para apoiar a construção de objetos de aprendizagem. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2011.

GARCÍA, Cristian González *et al.* Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios. **Computer Networks**, v. 64, p. 143-158, 2014.

GARCÍA-MAGARIÑO, Iván; PALACIOS-NAVARRO, Guillermo. A model-driven approach for constructing ambient assisted-living multi-agent systems customized for Parkinson patients. **Journal of Systems and Software**, v. 111, p. 34-48, 2016.

GIEMZA, Adam; BOLLEN, Lars; HOPPE, H. Ulrich. LEMONADE: field-trip authoring and classroom reporting for integrated *mobile learning* scenarios with intelligent agent support. **International Journal of Mobile learning and Organization**, v. 5, n. 1, p. 96-114, 2011.

GÓMEZ, S. Learning Design Implementation in Context-Aware and Adaptive Mobile Learning (Ph. D. Thesis). 2013.

GÓMEZ, Sergio *et al.* Context-aware adaptive and personalized mobile learning delivery supported by UoLmP. **Journal of King Saud University-Computer and Information Sciences**, v. 26, n. 1, p. 47-61, 2014.

GREAt. Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREAt). Disponível em: <http://great.ufc.br/>. Acesso em: 20 de janeiro 2016.

GRANT, Wayne C. Wireless coyote: A computer-supported field trip. **Commun. ACM**, v. 36, n. 5, p. 57-59, 1993.

GUO, Bin; ZHANG, Daqing; IMAI, Michita. Toward a cooperative programming framework for context-aware applications. **Personal and ubiquitous computing**, v. 15, n. 3, p. 221-233, 2011.

HAILPERN, Brent; TARR, Peri. Model-driven development: The good, the bad, and the ugly. **IBM systems journal**, v. 45, n. 3, p. 451-461, 2006.

HERMIDA, Jesús M. *et al.* Applying model-driven engineering to the development of Rich Internet Applications for Business Intelligence. **Information Systems Frontiers**, v. 15, n. 3, p. 411-431, 2013.

HOPPE, H. Ulrich; GAßNER, Katrin. Integrating collaborative concept mapping tools with group memory and retrieval functions. In: *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*. International Society of the Learning Sciences, 2002. p. 716-725

HWANG, Gwo-Jen; TSAI, Chin-Chung; YANG, Stephen JH. Criteria, strategies and research issues of context-aware *ubiquitous learning*. **Journal of Educational Technology & Society**, v. 11, n. 2, p. 81-91, 2008.

HUNG, Pi-Hsia *et al.* Seamless connection between learning and assessment-applying progressive learning tasks in mobile ecology inquiry. **Journal of Educational Technology & Society**, v. 16, n. 1, p. 194-205, 2013.

IMS GLOBAL LEARNING CONSORTIUM, Inc *et al.* IMS learning design information model. **IMS LD version v. 1**, 2003. Disponível em: http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html. Último acesso em 15 de maio de 2015.

IVANOV, Rosen. NFC-based pervasive learning service for children. In: **Proceedings of the 14th International Conference on Computer Systems and Technologies**. ACM, 2013. p. 329-336.

JEONG, Chang-Won; JOO, Su-Chong; JEONG, Young Sik. Sleeping situation monitoring system in ubiquitous environments. **Personal and ubiquitous computing**, v. 17, n. 7, p. 1357-1364, 2013.

JOUAULT, Frédéric *et al.* ATL: A model transformation tool. **Science of computer programming**, v. 72, n. 1, p. 31-39, 2008.

KELLY, Steven; TOLVANEN, Juha-Pekka. **Domain-specific modeling: enabling full code generation**. John Wiley & Sons, 2008.

KINDBERG, Tim; FOX, Armando. System software for ubiquitous computing. **IEEE pervasive computing**, v. 1, n. 1, p. 70-81, 2002.

Koper, Rob; Miao, Yongwu. (2007). Using the IMS LD standard to describe learning designs. Disponível em <http://lnx-hrl-075v.web.pwo.ou.nl/handle/1820/927>. Último acesso em 15 de maio de 2015.

KORTUEM, Gerd *et al.* Smart objects as building blocks for the internet of things. **Internet Computing, IEEE**, v. 14, n. 1, p. 44-51, 2010.

KOPER, Rob; MIAO, Yongwu. Using the IMS LD standard to describe learning designs. 2007. Disponível em <http://lnx-hrl-075v.web.pwo.ou.nl/handle/1820/927>. Último acesso em 15 de maio de 2015.

KRAVCIK, Milos *et al.* Mobile collector for field trips. **Journal of educational technology & society**, v. 7, n. 2, p. 25-33, 2004.

LEE, JongSuk Ruth *et al.* A ubiquitous smart learning platform for the 21st smart learners in an advanced science and engineering education. In: **Network-Based Information Systems (NBIS), 2012 15th International Conference on**. IEEE, 2012. p. 733-738.

LEWIS, James R.; SAURO, Jeff. The factor structure of the system usability scale. In: **Human Centered Design**. Springer Berlin Heidelberg, 2009. p. 94-103.

LIMA, Edmilson *et al.* GREAt Tour: Um Guia de Visitas Móvel e Sensível ao Contexto. In: **XII Workshop on Tools and Applications, 19th Brazilian Symposium on Multimedia and the Web**, Brasil, 2013.

LIMA, Luciana de; MARÇAL, Edgar; RIBEIRO, Júlio; ANDRADE, Rossana; VIANA, Windson, e MELO JÚNIOR, Antonio. A Guide for the Development and Use of *M-learning* Applications in Mathematics. **IEEE Technology and Engineering Education (ITEE)**, v. 6, n. 2, p. 1-12, 2011.

LO, Wan-Tzu; QUINTANA, Chris. Students' use of mobile technology to collect data in guided inquiry on field trips. In: **Proceedings of the 12th International Conference on Interaction Design and Children**. ACM, 2013. p. 297-300.

MACHADO, C. *et al.* Architectural elements of ubiquitous systems: A systematic review. In: **ICSEA 2013, The Eighth International Conference on Software Engineering Advances**. 2013. p. 208-213.

MACIEL, Cristiano *et al.* A Multi-agent Architecture to Support Ubiquitous Applications in Smart Environments. In: **Agent Technology for Intelligent Mobile Services and Smart**

Societies. Springer Berlin Heidelberg, 2015. p. 106-116.

MADANI, Halima Hebiri *et al.* Towards accessible and personalized *mobile learning* for learners with disabilities. In: **Information and Communication Technology and Accessibility (ICTA), 2013 Fourth International Conference on**. IEEE, 2013. p. 1-6.

MARÇAL, Edgar; ANDRADE, Rossana; RIOS, Riverson. Aprendizagem utilizando dispositivos móveis com sistemas de realidade virtual. **RENOTE**, v. 3, n. 1, 2005.

MARÇAL, Edgar *et al.* Geomóvel: Um Aplicativo para Auxílio a Aulas de Campo de Geologia. In: Anais do Simpósio Brasileiro de Informática na Educação. 2013.

MARÇAL, Edgar *et al.* A *mobile learning* system to enhance field trips in geology. In: Frontiers in Education Conference (FIE), 2014 IEEE. IEEE, p. 1-8, 2014.

MARINHO, Fabiana G. *et al.* MobiLine: A Nested Software Product Line for the domain of mobile and context-aware applications. **Science of Computer Programming**, v. 78, n. 12, p. 2381-2398, 2013.

MARTÍNEZ-ORTIZ, Iván; SIERRA, José Luis; FERNÁNDEZ-MANJÓN, Baltasar. Enhancing IMS LD Units of Learning Comprehension. In: **Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on**. IEEE, 2009. p. 561-566.

MAVROUDI, Anna; HADZILACOS, Thanasis. Implementation of adaptive learning designs. **Annals of the University of Craiova, Series: Automation, Computers, Electronics and Mechatronics**, v. 9, n. 2, 2012.

MAYER, Richard E. Multimedia learning. **Psychology of learning and motivation**, v. 41, p. 85-139, 2002.

MEDZINI, Arnon; MEISHAR-TAL, Hagit; SNEH, Yael. Use of mobile technologies as support tools for geography field trips. **International Research in Geographical and Environmental Education**, v. 24, n. 1, p. 13-23, 2015.

MEEK, S., FITZGERALD, E., PRIESTNALL, G., SHARPLES, M. Field Trip Learning. In: **Kinuthia, W. & Marshall, S. (eds). On the Move: Mobile Learning for Development. Information Age Publishing Inc.**, Charlotte, NC, 2013.

MERCADAL, Julien *et al.* A domain-specific approach to architecting error handling in pervasive computing. **ACM Sigplan Notices**, v. 45, n. 10, p. 47-61, 2010.

MEDEIROS, Flávio Mota; DE ALMEIDA, Eduardo Santana; DE LEMOS MEIRA, Silvio Romero. Towards an approach for service-oriented product line architectures. In: **Proceedings**

of the **Workshop on Service-oriented Architectures and Software Product Lines**. 2009. p. 1-7.

MIT, Massachusetts Institute of Technology. **App Inventor**. 2012. Disponível em: <http://appinventor.mit.edu/>.

MOHAGHEGHI, Parastoo et al. MDE adoption in industry: challenges and success criteria. In: **Models in Software Engineering**. Springer Berlin Heidelberg, 2009. p. 54-59.

MORAIS, Nídia Salomé et al. Uma Revisão de Literatura sobre o Uso das Tecnologias da Comunicação no Ensino Superior. **Revista PRISMA.COM**, n. 24, 2014.

MOURA, César. **Conceiving and Implementing a language-oriented approach for the design of automated learning scenarios**. 2007. Tese de Doutorado. Université des Sciences et Technologie de Lille-Lille I.

MUGWANYA, Raymond; MARSDEN, Gary. *Mobile learning* content authoring tools (MLCATs): a systematic review. In: **E-Infrastructures and E-Services on Developing Countries**. Springer Berlin Heidelberg, 2010. p. 20-31.

MUK, Alexander; CHUNG, Christina. Applying the technology acceptance model in a two-country study of SMS advertising. **Journal of Business Research**, v. 68, n. 1, p. 1-6, 2015.

MUNNELLY, Jennifer; FRITSCH, Serena; CLARKE, Siobhan. An aspect-oriented approach to the modularisation of context. In: **Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on**. IEEE, 2007. p. 114-124.

MYERS, Bruce Leigh; SCHROEDER, Trevor. An Application of the Technology Acceptance Model to Intended Adoption of Digital Printing Technology in the Label Industry. **Refereed Articles**, p. 10, 2014.

NABORS, Martha L.; EDWARDS, Linda Carol; MURRAY, R. Kent. Making the case for field trips: What research tells us and what site coordinators have to say. **Education**, v. 129, n. 4, p. 661-667, 2009.

NIELSEN, Jakob; MOLICH, Rolf. Heuristic evaluation of user interfaces. In: **Proceedings of the SIGCHI conference on Human factors in computing systems**. ACM, 1990. p. 249-256.

NIEUWDORP, Eva. The pervasive discourse: an analysis. **Computers in Entertainment (CIE)**, v. 5, n. 2, p. 13, 2007.

OGATA, Hiroaki; YANO, Yoneo. Context-aware support for computer-supported *ubiquitous learning*. In: **Wireless and Mobile Technologies in Education, 2004. Proceedings. the 2nd**

IEEE International Workshop on. IEEE, 2004. p. 27-34.

OKERLUND, Johanna; TURBAK, Franklyn. A Preliminary Analysis of App Inventor Blocks Programs. **Poster presented at Visual Languages and Human Centric Computing (VLHCC), Sept.** p. 15-19, 2013.

OLIVEIRA, Francisco Hélio; SALVADOR, Laís; NOVAIS, Renato. Uma Análise do Uso da Ontologia IMS LD na Construção de Modelos Conceituais para E-learning. In: **Anais do Simpósio Brasileiro de Informática na Educação.** 2014. p. 1213-1222.

OLSEN, Gøran K.; OLDEVIK, Jon. Scenarios of traceability in model to text transformations. In: **Model Driven Architecture-Foundations and Applications.** Springer Berlin Heidelberg, 2007. p. 144-156.

PARK, Eunil et al. Determinants of player acceptance of mobile social network games: An application of extended technology acceptance model. **Telematics and Informatics**, v. 31, n. 1, p. 3-15, 2014.

PENG, Hsinyi *et al.* Ubiquitous knowledge construction: *Mobile learning* re-defined and a conceptual framework. **Innovations in Education and Teaching International**, v. 46, n. 2, p. 171-183, 2009.

PENG, Hsinyi *et al.* Ubiquitous knowledge construction: *Mobile learning* re-defined and a conceptual framework. **Innovations in Education and Teaching International**, v. 46, n. 2, p. 171-183, 2009.

QUINN, Thomas P. Turning Class Field Trips into Long-Term Research: A Great Idea with a Few Pitfalls. **Fisheries**, v. 40, n. 2, p. 65-68, 2015.

PEREIRA, Fernanda. **Avaliação de usabilidade em bibliotecas digitais: um estudo de caso.** 2011. 122 f. 2011. Dissertação de Mestrado em Ciência da Informação – Escola de Ciência da Informação, Universidade Federal de Minas Gerais, Belo Horizonte.

PETERSEN, Kai et al. Systematic mapping studies in software engineering. In: **12th International Conference on Evaluation and Assessment in Software Engineering.** sn, 2008.

PREUVENEERS, Davy; NOVAIS, Paulo. A survey of software engineering best practices for the development of smart applications in Ambient Intelligence. **Journal of Ambient Intelligence and Smart Environments**, v. 4, n. 3, p. 149-162, 2012.

RATNA, P. A.; MEHRA, Saloni. Exploring the acceptance for e - learning using technology acceptance model among university students in India. **International Journal of Process Management and Benchmarking**, v. 5, n. 2, p. 194-210, 2015.

REEVES, Leah M. *et al.* Guidelines for multimodal user interface design. **Communications of the ACM**, v. 47, n. 1, p. 57-59, 2004.

ŘEŘUCHA, Šimon *et al.* The BAARA (Biological AutomAted RAdiotracking) System: A New Approach in Ecological Field Studies. **PloS one**, v. 10, n. 2, p. e0116785, 2015.

ROCHA, L. S. (2013). CAEHv: Um Método para Verificação de Modelos do Tratamento de Exceção Sensível ao Contexto em Sistemas Ubíquos. **Tese submetida ao programa de Mestrado e Doutorado em Ciência da Computação**, da Universidade Federal do Ceará como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ROSLIN, Ma Anya Yasmin A. *et al.* Social impact of ecotourism on the behavior of students on educational field trips to Makiling Botanic Gardens in the University of the Philippines Los Baños. **USM R&D Journal**, v. 17, n. 1, p. 71-80, 2009.

RUNESON, Per; HÖST, Martin. Guidelines for conducting and reporting case study research in software engineering. **Empirical software engineering**, v. 14, n. 2, p. 131-164, 2009.

SALMAN, Abdulrahman; POURABDOLLAH, Amir. UCML: Ubiquitous Control Modeling Language for Android Platform. In: **Next Generation Mobile Applications, Services and Technologies (NGMAST), 2010 Fourth International Conference on**. IEEE, 2010. p. 112-116.

SÁNCHEZ, Jaime; OLIVARES, Ruby. Problem solving and collaboration using mobile serious games. **Computers & Education**, v. 57, n. 3, p. 1943-1952, 2011.

SANTOS, Rainara M. *et al.* A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems. In: **Human Computer Interaction**. Springer International Publishing, 2013. p. 63-70.

SAURO, Jeff. **A practical guide to the system usability scale: Background, benchmarks & best practices**. Measuring Usability LLC, 2011.

SAURO, Jeff; LEWIS, James R. **Quantifying the user experience: Practical statistics for user research**. Elsevier, 2012.

SCHMITZ, Birgit *et al.* Designing a mobile learning game to investigate the impact of role-playing on helping behaviour. In: **Scaling up Learning for Sustained Impact**. Springer Berlin Heidelberg, 2013. p. 357-370.

SERRAL, Estefanía; VALDERAS, Pedro; PELECHANO, Vicente. Towards the model driven development of context-aware pervasive systems. **Pervasive and Mobile Computing**, v. 6, n. 2, p. 254-280, 2010.

SHAKIL A., FAIZI W. E HAFEEZ S. **The need and importance of field trips at higher level in karachi, pakistan.** In: International Journal of Academic Research in Business and Social Sciences, June, Vol. 2, No. 1, 2011.

SHARPLES, Mike; ROSCHELLE, Jeremy. Guest editorial: Special section on mobile and ubiquitous technologies for learning. **IEEE Transactions on Learning Technologies**, n. 1, p. 4-6, 2010.

SNOW, Kellie et al. Considering the user perspective: Research into usage and communication of digital information. **D-Lib Magazine**, v. 14, n. 5, p. 3, 2008.

SPÍNOLA, Rodrigo Oliveira; TRAVASSOS, Guilherme Horta. Towards a framework to characterize ubiquitous software projects. **Information and Software Technology**, v. 54, n. 7, p. 759-785, 2012.

STEFAN, Livia *et al.* *Ubiquitous learning* Solutions for Remote Communities--A Case Study for K-12 Classes in a Romanian Village. In: **Control Systems and Computer Science (CSCS), 2013 19th International Conference on.** IEEE, 2013. p. 569-574.

SUN, Qingquan et al. A multi-agent-based intelligent sensor and actuator network design for smart house and home automation. **Journal of Sensor and Actuator Networks**, v. 2, n. 3, p. 557-588, 2013.

TABUENCA, Bernardo *et al.* Mobile authoring of open educational resources for authentic learning scenarios. **Universal Access in the Information Society**, p. 1-15, 2014.

TERNIER, Stefaan *et al.* ARLearn: Augmented Reality Meets Augmented Virtuality. **J. UCS**, v. 18, n. 15, p. 2143-2164, 2012.

TESORIERO, Ricardo *et al.* CAUCE: Model-driven Development of Context-aware Applications for Ubiquitous Computing Environments. **J. UCS**, v. 16, n. 15, p. 2111-2138, 2010.

THÜM, Thomas *et al.* A classification and survey of analysis strategies for software product lines. **ACM Computing Surveys (CSUR)**, v. 47, n. 1, p. 6, 2014.

V VIEIRA, Paulo; COSTA, Agnaldo; LA RAABE, André. Um Estudo Comparativo sobre a Utilização dos Padrões Common Cartridge e SCORM no Ambiente Moodle. **Conferencias LACLO**, v. 3, n. 1, 2012.

VÖLTER, Markus *et al.* **Model-driven software development: technology, engineering, management.** John Wiley & Sons, 2013.

WANG, Shu-Lin; WU, Chun-Yi. Application of context-aware and personalized recommendation to implement an adaptive *ubiquitous learning* system. **Expert Systems with applications**, v. 38, n. 9, p. 10831-10838, 2011.

WASSERMAN, Anthony I. Software engineering issues for mobile application development. In: **Proceedings of the FSE/SDP workshop on Future of software engineering research**. ACM, 2010. p. 397-400.

WAZLAWICK, Raul. Metodologia de pesquisa para ciência da computação. Elsevier Brasil, 2009.

WEISER, Mark. The computer for the 21st century. **Scientific american**, v. 265, n. 3, p. 94-104, 1991.

WENG, Wei Kai; CHANG, Polun. Developing Mobile Support System for Dynamic Integrated Community-Based Screening. In: Nursing Informatics 2014: East Meets West ESMART. Proceedings of the 12th International Congress on Nursing Informatics, Taipei, Taiwan, June 21-25, 2014. **IOS Press**, 2014. p. 257.

WHITTLE, Jon et al. Industrial adoption of model-driven engineering: are the tools really the problem?. In: **Model-Driven Engineering Languages and Systems**. Springer Berlin Heidelberg, 2013. p. 1-17.

WILLIAMS, James R. *et al.* **Sensitivity analysis in model-driven engineering**. Springer Berlin Heidelberg, 2012.

WOLBER, David. App inventor and real-world motivation. In: **Proceedings of the 42nd ACM technical symposium on Computer science education**. ACM, 2011. p. 601-606.

WU, Po-Han; HWANG, Gwo-Jen; TSAI, Wen-Hung. An expert system-based context-aware *ubiquitous learning* approach for conducting science learning activities. **Journal of Educational Technology & Society**, v. 16, n. 4, p. 217-230, 2013.

ZBICK, Janosch et al. A web-based framework to design and deploy mobile learning activities: Evaluating its usability, learnability and acceptance. In: **Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on**. IEEE, 2015. p. 88-92.

ZERVAS, Panagiotis; SAMPSON, Demetrios G. Supporting Mobile Access to Online Courses: The ASK Mobile SCORM Player and the ASK Mobile LD Player. **Increasing Access**, p. 125, 2014.

APÊNDICE A – Listagem das Referências dos 63 Artigos Analisados no Mapeamento Sistemático

1	AHMED, Sohaib; PARSONS, David. Abductive science inquiry using mobile devices in the classroom. Computers & Education , v. 63, p. 62-72, 2013.
2	AIZAWA, Kiyoharu et al. Location identification for visitor behavior log in museum. In: Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry . ACM, 2010. p. 369-372.
3	ATIF, Yacine. Learning services provisioning using semantic web technologies. In: Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications . ACM, 2010. p. 24.
4	BAI, Xin. Affordance of <i>ubiquitous learning</i> through cloud computing. In: Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference on . IEEE, 2010. p. 78-82.
5	BAI, Xin. Affordance of <i>ubiquitous learning</i> through cloud computing. In: Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference on . IEEE, 2010. p. 78-82.
6	BALOIAN, Nelson et al. Learning with patterns: An effective way to implement computer supported pervasive learning. In: Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference on . IEEE, 2010. p. 677-682.
7	CHANG, Cheng-Sian; CHEN, Tzung-Shi; HSU, Wei-Hsiang. The study on integrating WebQuest with <i>mobile learning</i> for environmental education. Computers & Education , v. 57, n. 1, p. 1228-1239, 2011.
8	CHANG, Kuo-En et al. Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum. Computers & Education , v. 71, p. 185-197, 2014.
9	CHANG, Wan-Jen; YEH, Zong-Mu; CHENG, Kin-Jung. Developing a <i>mobile learning</i> system to community-based learning for rural elementary school students. In: Computer Symposium (ICS), 2010 International . IEEE, 2010. p. 451-456.
10	CHEN, Chia-Chen; HUANG, Tien-Chi. Learning in a u-Museum: Developing a context-aware <i>ubiquitous learning</i> environment. Computers & Education , v. 59, n. 3, p. 873-883, 2012.
11	CHEN, Nian-Shing et al. Augmenting paper-based reading activity with direct access to digital materials and scaffolded questioning. Computers & Education , v. 57, n. 2, p. 1705-1715, 2011.
12	CHIANG, Tosti HC; YANG, Stephen JH; HWANG, Gwo-Jen. Students' online interactive patterns in augmented reality-based inquiry activities. Computers & Education , v. 78, p. 97-108, 2014.
13	CHIOU, Chuang-Kai; TSENG, Judy CR. Design of a personalized navigation support system for context-aware <i>ubiquitous learning</i> environment. In: Proceedings of the 2012 RecSys workshop on Personalizing the local mobile experience . ACM, 2012. p. 1-6.
14	CHU, Hui-Chun et al. A two-tier test approach to developing location-aware <i>mobile learning</i> systems for natural science courses. Computers & Education , v. 55, n. 4, p. 1618-1627, 2010.
15	CHU, Hui-Chun; CHEN, Shun-Hua. Using WebQuests Learning Strategy in Context-Aware <i>Ubiquitous learning</i> Environment for English Course. In: Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference on . IEEE, 2014. p.

	547-548.
16	CHU, Hui-Chun; HWANG, Gwo-Jen. A location-aware <i>mobile learning</i> system to provide field learning guidance for natural science courses. In: Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on . IEEE, 2010. p. 291-294.
17	CHU, Hui-Chun; LIN, Chia-Wei. The Development and Application of a Repertory Grid-Oriented Ubiquitous Augmented Reality Learning System. In: Advanced Applied Informatics (IIAIAI), 2013 IIAI International Conference on . IEEE, 2013. p. 207-210.
18	ELLIS, Kirsten; RAY, Neil; HOWARD, Cheryl. Learning a physical skill via a computer: a case study exploring Australian Sign Language. In: Proceedings of the 23rd Australian Computer-Human Interaction Conference . ACM, 2011. p. 98-103.
19	FERNÁNDEZ-LÓPEZ, Álvaro et al. <i>Mobile learning</i> technology based on iOS devices to support students with special education needs. Computers & Education , v. 61, p. 77-90, 2013.
20	GICQUEL, Pierre-Yves; LENNE, Dominique; MOULIN, Claude. Design and use of CALM: An ubiquitous environment for <i>mobile learning</i> during museum visit. In: Digital Heritage International Congress (DigitalHeritage), 2013 . IEEE, 2013. p. 645-652.
21	GÓMEZ GÓMEZ, Jorge; HUETE, Juan F.; HERNANDEZ RIAÑO, Velssy. Learning system based on contextual awareness for clinical practice in nursing courses. In: Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on . IEEE, 2014. p. 186-190.
22	GÓMEZ, Sergio et al. Context-aware adaptive and personalized <i>mobile learning</i> delivery supported by UoLmP. Journal of King Saud University-Computer and Information Sciences , v. 26, n. 1, p. 47-61, 2014.
23	GUAZZARONI, Giuliana. Emotional mapping of the archaeologist game. Computers in Human Behavior , v. 29, n. 2, p. 335-344, 2013.
24	HONG, Jon-Chao et al. Using calibration to enhance students' self-confidence in English vocabulary learning relevant to their judgment of over-confidence and predicted by <i>smartphone</i> self-efficacy and English learning anxiety. Computers & Education , v. 72, p. 313-322, 2014.
25	HSIEH, Sheng-Wen et al. Effects of teaching and learning styles on students' reflection levels for <i>ubiquitous learning</i> . Computers & Education , v. 57, n. 1, p. 1194-1201, 2011.
26	HUANG, Hui-Wen; WU, Chih-Wei; CHEN, Nian-Shing. The effectiveness of using procedural scaffoldings in a paper-plus- <i>smartphone</i> collaborative learning context. Computers & Education , v. 59, n. 2, p. 250-259, 2012.
27	HUANG, Yueh-Min et al. The design and implementation of a meaningful learning-based evaluation method for <i>ubiquitous learning</i> . Computers & Education , v. 57, n. 4, p. 2291-2302, 2011.
28	HUANG, Yueh-Min; LIN, Yen-Ting; CHENG, Shu-Chen. Effectiveness of a mobile plant learning system in a science curriculum in Taiwanese elementary education. Computers & Education , v. 54, n. 1, p. 47-58, 2010.

29	HWANG, Gwo-Haur et al. Development of a Web 2.0-based <i>ubiquitous learning</i> platform for schoolyard plant identification. In: Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on . IEEE, 2012. p. 259-263.
30	HWANG, Gwo-Jen; CHANG, Hsun-Fang. A formative assessment-based <i>mobile learning</i> approach to improving the learning attitudes and achievements of students. Computers & Education , v. 56, n. 4, p. 1023-1031, 2011.
31	HWANG, Gwo-Jen; WU, Po-Han; KE, Hui-Ru. An interactive concept map approach to supporting <i>mobile learning</i> activities for natural science courses. Computers & Education , v. 57, n. 4, p. 2272-2280, 2011.
32	IACOVIDES, Ioanna; COX, Anna L.; BLANDFORD, Ann. Supporting learning within the workplace: Device training in healthcare. In: Proceedings of the 31st European Conference on Cognitive Ergonomics . ACM, 2013. p. 30.
33	IKEDA, Aya; ARAI, Toshifumi; SHIO, Itiro. Dream drill: learning application. In: CHI'12 Extended Abstracts on Human Factors in Computing Systems . ACM, 2012. p. 1853-1858.
34	IVANOV, Rosen. TeachMe: personal learning environment for children. In: Proceedings of the 15th International Conference on Computer Systems and Technologies . ACM, 2014. p. 437-444.
35	JOO, Kil Hong; PARK, Nam Hun. Design and application of the u-SM teaching and learning model for an efficient <i>ubiquitous learning</i> . In: Computing, Management and Telecommunications (ComManTel), 2013 International Conference on . IEEE, 2013. p. 264-268.
36	LAI, Ah-Fur et al. Integrate Handheld Device and RFID to Support Context Awareness Environment for Outdoor Inquiry Learning Activity. In: Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), 2010 6th IEEE International Conference on . IEEE, 2010. p. 132-136.
37	LAINE, Teemu H. et al. Critical factors for technology integration in game-based pervasive learning spaces. Learning Technologies, IEEE Transactions on , v. 3, n. 4, p. 294-306, 2010.
38	LAINE, Teemu H.; DUVEKOG, Marcus; SUTINEN, Erkki. Bagamoyo caravan: pervasive learning game for a tanzanian museum. In: IST-Africa Conference Proceedings, 2011 . IEEE, 2011. p. 1-8.
39	LIU, Tsung-Yu; CHU, Yu-Ling. Using ubiquitous games in an English listening and speaking course: Impact on learning outcomes and motivation. Computers & Education , v. 55, n. 2, p. 630-643, 2010.
40	LIU, Tzu-Chien; LIN, Yi-Chun; PAAS, Fred. Effects of prior knowledge on learning from different compositions of representations in a <i>mobile learning</i> environment. Computers & Education , v. 72, p. 328-338, 2014.
41	MANDULA, Kumar; MEDA, Srinivas Rao; JAIN, Dhanander Kumar. Research and implementation of a mobile video streaming application for <i>ubiquitous learning</i> . In: Technology Enhanced Education (ICTEE), 2012 IEEE International Conference on . IEEE, 2012. p. 1-6.
42	MARTIN, Florence; ERTZBERGER, Jeffrey. Here and now <i>mobile learning</i> : An experimental study on the use of mobile technology. Computers & Education , v. 68, p. 76-85, 2013.
43	MUNOZ-CRISTOBAL, Juan et al. Supporting Teacher Orchestration in <i>Ubiquitous learning</i> Environments: A Study in Primary Education. Learning Technologies, IEEE Transactions on , v. 8, n. 1, p. 83-97, 2015.

44	PERSADA, Satria Fadil et al. Toward Paperless Public Announcement on Environmental Impact Assessment (EIA) through SMS Gateway in Indonesia. Procedia Environmental Sciences , v. 20, p. 271-279, 2014.
45	POURSAEED, Bahar; LEE, Chien-Sing. Self-initiated curriculum planning, visualization and assessment in improving meaningful learning: A comparison between mobile and <i>ubiquitous learning</i> . In: Technology for Education (T4E), 2010 International Conference on . IEEE, 2010. p. 107-113.
46	RADU, Iulian. Augmented reality in education: a meta-review and cross-media analysis. Personal and Ubiquitous Computing , v. 18, n. 6, p. 1533-1543, 2014.
47	REDONDO, Ernest et al. New strategies using handheld augmented reality and <i>mobile learning</i> -teaching methodologies, in architecture and building engineering degrees. Procedia Computer Science , v. 25, p. 52-61, 2013.
48	RIAZA, Blanca García; RODRÍGUEZ, Ana Iglesias. Attitude towards the use of mobile devices for the practice of oral skills in English. In: Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality . ACM, 2014. p. 387-391.
49	SÁNCHEZ, Iván et al. NFC-based interactive learning environments for children. In: Proceedings of the 10th international conference on interaction design and children . ACM, 2011. p. 205-208.
50	SÁNCHEZ, Jaime; OLIVARES, Ruby. Problem solving and collaboration using mobile serious games. Computers & Education , v. 57, n. 3, p. 1943-1952, 2011.
51	SANTOS, Patricia et al. QuesTInSitu: From tests to routes for assessment in situ activities. Computers & Education , v. 57, n. 4, p. 2517-2534, 2011.
52	SCOTT, Kristopher; BENLAMRI, Rachid. Context-aware services for smart learning spaces. Learning Technologies, IEEE Transactions on , v. 3, n. 3, p. 214-227, 2010.
53	SONG, Yanjie. "Bring Your Own Device (BYOD)" for seamless science inquiry in a primary school. Computers & Education , v. 74, p. 50-60, 2014.
54	STEFAN, Livia et al. <i>Ubiquitous learning</i> Solutions for Remote Communities--A Case Study for K-12 Classes in a Romanian Village. In: Control Systems and Computer Science (CSCS), 2013 19th International Conference on . IEEE, 2013. p. 569-574.
55	SU, Chung-Ho; CHENG, Ching-Hsue. A Mobile Game-based Insect Learning System for improving the learning achievements. Procedia-Social and Behavioral Sciences , v. 103, p. 42-50, 2013.
56	SUNG, Han-Yu et al. A prompt-based annotation approach to conducting <i>mobile learning</i> activities for architecture design courses. Computers & Education , v. 76, p. 80-90, 2014.
57	TSENG, Judy CR; WU, Chih-Hsiang; HWANG, Gwo-Jen. A collaborative <i>ubiquitous learning</i> approach for conducting personal computer-assembling activities. In: Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on . IEEE, 2010. p. 726-727.
58	WANG, Chun-Chia; WEI, Ching-Ren. The Implementation of a Context-Aware Mobile Japanese Conversation Learning System Based on NFC-enabled <i>Smartphones</i> . In: Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference on . IEEE, 2014. p. 313-317.

59	YANG, Xiu-Jun et al. Effects of Video-Based Reflection Prompts on Learners' Reflection Levels in a Context-Aware <i>U-learning</i> Environment. In: Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on. IEEE, 2012. p. 263-267.
60	YE, Shu-Hao; HUNG, Yen-Chu. The study of self-seamless teaching strategy for <i>ubiquitous learning</i> environments. In: Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), 2010 6th IEEE International Conference on. IEEE, 2010. p. 182-186.
61	ZBICK, Janosch et al. mLearn4web: a web-based framework to design and deploy cross-platform mobile applications. In: Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2014. p. 252-255.
62	ZHANG, Jia et al. The development and evaluation of an augmented reality-based armillary sphere for astronomical observation instruction. Computers & Education , v. 73, p. 178-188, 2014.
63	ZHAO, Xinyou; WAN, Xin; OKAMOTO, Tatsuaki. Adaptive content delivery in <i>ubiquitous learning</i> environment. In: Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), 2010 6th IEEE International Conference on. IEEE, 2010. p. 19-26.

APÊNDICE B – Instrumentos de Avaliação

1º) Questionário de Avaliação – Usabilidade do UFC-Inventor

PARTE I – Sobre o UFC-Inventor.

1) Eu acho que gostaria de usar este sistema frequentemente.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

2) Eu achei o sistema desnecessariamente complexo.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

3) Eu achei o sistema fácil para usar.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

4) Eu acho que precisaria do apoio de um suporte técnico para ser possível usar este sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

5) Eu achei que as diversas funções neste sistema foram bem integradas.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

6) Eu achei que houve muita inconsistência neste sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

7) Eu imaginaria que a maioria das pessoas aprenderia a usar esse sistema rapidamente.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

8) Eu achei o sistema muito pesado para uso.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

9) Eu me senti muito confiante em utilizar esse sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

10) Eu precisei aprender uma série de coisas antes que eu pudesse começar a utilizar esse sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

PARTE II – Questões abertas

11) Quais os pontos positivos, negativos e sugestões sobre o UFC-Inventor você destaca?

2º) Questionário de Avaliação – Aplicação “Aula de Campo”

Modelo do Celular/SO: _____

Você já usou outro aplicativo educativo de celular em campo anteriormente: () NÃO () SIM

PARTE I – Sobre a aplicação usada em campo.

1) Eu acho que gostaria de usar este sistema frequentemente.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

2) Eu achei o sistema desnecessariamente complexo.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

3) Eu achei o sistema fácil para usar.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

4) Eu acho que precisaria do apoio de um suporte técnico para ser possível usar este sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

5) Eu achei que as diversas funções neste sistema foram bem integradas.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

6) Eu achei que houve muita inconsistência neste sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

7) Eu imaginaria que a maioria das pessoas aprenderia a usar esse sistema rapidamente.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

8) Eu achei o sistema muito pesado para uso.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

9) Eu me senti muito confiante em utilizar esse sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

10) Eu precisei aprender uma série de coisas antes que eu pudesse começar a utilizar esse sistema.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

PARTE II – A aplicação ...

11) Facilitou-me coletar os dados usando recursos como a câmera e o microfone.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

12) Dificultou-me a escrever textos e números.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

13) Permitiu-me executar as tarefas em campo mais eficientemente.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

14) Complicou a minha execução das tarefas de aprendizagem.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

15) É uma tecnologia útil para coleta dos dados.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

16) Ajudou-me a compreender melhor os conceitos da aula de campo.

() Discordo Totalmente () Discordo () Indiferente () Concordo () Concordo Totalmente

PARTE III – Questões abertas

17) Como você preferiu responder as perguntas na aplicação?

() Texto () Áudio () Foto. Por quê?

18) Quais os pontos positivos, negativos e sugestões sobre a aplicação que você usou?

APÊNDICE C – Respostas dos Alunos à Primeira Parte do Questionário de Avaliação

											SUS	Usabilidade	Capacidade de Aprendizado
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10			
Aluno 1	5	1	5	1	4	1	5	1	4	2	92,5	93,8	87,5
Aluno 2	5	2	5	2	4	2	5	1	4	2	85,0	87,5	75,0
Aluno 3	4	1	5	2	4	2	4	1	4	2	82,5	84,4	75,0
Aluno 4	4	2	4	3	3	2	4	2	4	2	70,0	71,9	62,5
Aluno 5	4	2	4	2	4	2	5	1	5	2	82,5	84,4	75,0
Aluno 6	4	1	5	1	4	1	5	2	4	1	90,0	87,5	100,0
Aluno 7	5	1	5	2	4	2	5	1	4	3	85,0	90,6	62,5
Aluno 8	4	1	5	2	4	1	5	1	4	1	90,0	90,6	87,5
Aluno 9	4	2	4	2	4	2	3	2	3	2	70,0	68,8	75,0
Aluno 10	5	1	5	1	4	2	4	2	4	1	87,5	84,4	100,0
Aluno 11	4	2	4	2	3	2	4	2	4	2	72,5	71,9	75,0
Aluno 12	3	2	4	2	4	2	4	2	4	2	72,5	71,9	75,0
Aluno 13	4	2	5	1	4	2	5	1	5	1	90,0	87,5	100,0
Aluno 14	5	1	5	2	4	2	5	1	4	2	87,5	90,6	75,0
Aluno 15	4	1	5	1	5	2	5	1	5	2	92,5	93,8	87,5
Aluno 16	3	2	5	2	3	2	4	1	4	2	75,0	75,0	75,0
Aluno 17	4	1	5	1	4	2	5	1	5	1	92,5	90,6	100,0
Médias											83,4	83,8	81,6