



UNIVERSIDADE FEDERAL DO CEARÁ  
CENTRO DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

MARIA DE FÁTIMA COSTA DE SOUZA

*Customização Guiada:*  
Uma Estratégia Orientada a Modelos para a Produção de  
Objetos de Aprendizagem

FORTALEZA

2012

MARIA DE FÁTIMA COSTA DE SOUZA

*Customização Guiada:*

Uma Estratégia Orientada a Modelos para a Produção de Objetos de  
Aprendizagem

Tese apresentada ao Programa de Pós-graduação  
em Engenharia de Teleinformática da UFC, como  
parte dos requisitos para obtenção do título de  
DOUTOR(A) em Engenharia de Teleinformá-  
tica. Área de Concentração: Sinais e Sistemas.

**Orientadora: Profa. Rossana Maria de Castro Andrade**

**PhD em Ciência da Computação**

**Coorientador: Prof. José Aires de Castro Filho**

**PhD em Educação Matemática**

FORTALEZA

2012

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Pós-graduação em Engenharia

---

S716c

SOUZA, MARIA DE FÁTIMA COSTA DE.

Customização guiada: uma estratégia orientada a modelos para produção de objetos de aprendizagem - 2012

256 f. : il. color., enc. ; 30 cm.

Tese (doutorado) - Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia de Teleinformática, Programa de Pós - Graduação em Engenharia de Teleinformática, Fortaleza, 2012.

Área de Concentração: Sinais e Sistemas.

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

Coorientação: Prof. Dr. José Aires de Castro Filho.

1. Teleinformática. 2. Engenharia de software. I. Título.

CDD 621.38

---

MARIA DE FÁTIMA COSTA DE SOUZA

*Customização Guiada:*

Uma Estratégia Orientada a Modelos para a Produção de Objetos de  
Aprendizagem

Tese apresentada ao Programa de Pós-graduação  
em Engenharia de Teleinformática da UFC, como  
requisito parcial para a obtenção do grau de  
DOUTOR(A) em Engenharia de Teleinformá-  
tica.

Aprovado em 06 de Março de 2012

**BANCA EXAMINADORA**

---

Profa. Rossana Maria de Castro Andrade  
Universidade Federal do Ceará (Orientadora)

---

Prof. José Aires de Castro Filho  
Universidade Federal do Ceará (Co-orientador)

---

Prof. Giovanni Cordeiro Barroso  
Universidade Federal do Ceará (Avaliador Interno)

---

Profa. Liane Margarida Rockenbach Tarouco  
Universidade Federal do Rio Grande do Sul (Avaliadora Externa)

---

Prof. Alex Sandro Gomes  
Universidade Federal de Pernambuco (Avaliador Externo)

*Aos meus pais, Expedito(In memorium) e Iraci (In memorium) que me ensinaram desde cedo, a importância do conhecimento. Ao meu esposo, Cidcley, que sempre acreditou em meu potencial e que faz das minhas conquistas as suas.*

## AGRADECIMENTOS

Apesar da construção de uma tese, ser na essência, um trabalho individual, há durante todo o seu período de criação, contribuições de natureza diversas que não podem e nem devem deixar de ser mencionados. Por essa razão, desejo expressar os meus sinceros agradecimentos:

À Professora Rossana, minha orientadora, pela competência científica e firmeza no acompanhamento deste trabalho, assim como pelas críticas, correções e sugestões relevantes feitas durante a orientação que contribuíram tanto para minha vida acadêmica quanto pessoal.

Ao Professor José Aires, meu co-orientador, pela competência científica, orientação dada, amizade e que, além disso, me ensinou com serenidade, prazer e dedicação parte do que sei sobre recursos educacionais.

Aos Professores Alex Sandro, Liane Tarouco e Giovanni Cordeiro pelas ricas contribuições ao meu trabalho. Me senti honrada com a participação de vocês em minha banca de doutorado.

Aos professores do Programa de Pós-Graduação em Engenharia de Teleinformática pelo grande aprendizado adquirido durante as disciplinas cursadas no doutorado, em especial aos professores Giovanni Cordeiro e Danielo Gomes.

Ao Prof. Charles Casimiro, coordenador do curso de Pós-Graduação em Engenharia de Teleinformática (PPGETI) por toda a dedicação despendida aos alunos deste programa.

Ao Renato, secretário do PPGETI, pela paciência, dedicação e prontidão em solucionar os problemas.

Ao Anchieta, aluno do curso de Engenharia de Computação do IFCE e bolsista que trabalhou comigo nesta pesquisa, por seu comprometimento e valiosa ajuda na implementação da ferramenta CLO Studio.

A todos os membros do Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (Proativa), em especial a Alisandra, Raquel, Juscileide,

Mixilene, Magno, Sibeles e Yuri por terem contribuído diretamente com a avaliação desta pesquisa

A todos que fazem ou fizeram parte do Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat) por toda atenção despendidos a mim, em especial à Darilu, Liliane, Christiane Carvalho e a Emanuela. Aos amigos Fabiana, Valéria, Smayle, Windson e Paulo Henrique pelas preciosas conversas, pois foram delas que extrai os valiosos insights deste trabalho.

A todos os meus queridos amigos e colegas da UFC Virtual que através de orações, pensamentos positivos, palavras de força e momentos de descontração, injetavam em mim o ânimo necessário para finalizar essa jornada. Em especial aos amigos Alisandra, Raquel, Priscila, Lincoln, Melo Jr, Glaudiney, Henrique Pequeno, Alexsandra e Ticianana.

Ao amigo e Professor do Instituto UFC Virtual, Herbert, por ter avisado e incentivado minha inscrição no processo de seleção do PPGETI.

Ao diretor do Instituto UFC Virtual, Prof. Mauro Cavalcante Pequeno, por estimular o crescimento acadêmico de todos os professores que fazem parte desta unidade acadêmica.

Aos meus queridos e amados pais, Expedito (In memorium) e Iraci (in memorium), por terem sido os primeiros anjos de minha vida e desde cedo buscaram com todo zelo alimentar em mim a perseverança, a determinação e a sede pelo conhecimento.

Aos meus irmãos, Netinho, Socorrinha, Liduína, Marquinhos, Franzé, Jeová e a minha querida prima Juscilene pelas orações e torcida. Um agradecimento especial, a meu irmão Marquinhos por ter me dado apoio no processo de revisão e tradução dos artigos em inglês.

Aos meus sogros, Cícero e Ivan e aos meus cunhados, cunhadas e sobrinhos pelo estímulo, torcida e orações.

Por fim, mas não menos importante, um agradecimento muito especial ao meu querido e amado esposo, Cidcley, que foi o maior incentivador para que eu ingressasse no mundo acadêmico. A você meu anjo, não somente agradeço como dedico esse trabalho.

## RESUMO

Objetos de Aprendizagem (OA) são recursos digitais utilizados tanto para auxiliar os professores em sala de aula na transmissão de conceitos como para facilitar a compreensão dos mesmos pelos alunos. No entanto, esses recursos possuem limitações que acabam por restringir os ganhos resultantes de sua utilização. Uma delas está relacionada a capacidade limitada de abertura desses recursos, fato que impede os professores, por si próprios, de adaptá-los às suas necessidades pedagógicas. Essa ideia de adaptação comunga com o desejo de compartilhamento e democratização de informação e conhecimento pregados pelo movimento dos recursos educacionais abertos (REA). No entanto, para efetuar as adaptações dentro da filosofia desse movimento, as estratégias existentes de abertura desses recursos requerem que os professores tenham acesso aos códigos fontes e conhecimento para realizar tais adaptações. Outro problema dos OA está relacionado aos processos de desenvolvimento empregados que não consideram a multidisciplinaridade das equipes que os produzem. No intuito de minimizar essas limitações, conceitos de Engenharia de Software são utilizados neste trabalho para propor uma estratégia de adaptação de recursos digitais, especificamente de OA, denominada de Customização Guiada. Essa estratégia tem por objetivo possibilitar a indicação, em tempo de projeto, dos elementos que podem ser manipulados pelos professores diretamente na interface dos recursos. Além disso, ao utilizar um processo de desenvolvimento de software orientado a modelos, que possui uma linguagem específica de domínio, a comunicação entre os membros da equipe é facilitada. Para implementar a estratégia proposta, uma ferramenta de autoria (CLO Studio) é desenvolvida e utilizada para avaliar tanto o desenvolvimento de um OA customizável (OAC), através da análise da manipulação do CLO Studio por uma equipe de desenvolvimento de OA, quanto a sua utilização, através da análise do uso desse tipo de recurso por professores. Os resultados mostram que o processo de desenvolvimento dos OAC tanto estimula a cooperação dos profissionais da equipe de desenvolvimento quanto facilita a comunicação entre eles. Além disso, a utilização desse tipo de recurso possibilita uma maior autonomia dos professores, abrindo novas oportunidades em suas práticas docentes.

**Palavras-Chave:** Adaptação de Software, Engenharia de Software Orientada a Modelos, Objetos de Aprendizagem (OA), Processo de Desenvolvimento, Recursos Educacionais Abertos (REA).



## ABSTRACT

Learning Objects (LO) are digital resources used to help teachers in the classroom as well as to support students to better understand their lessons. However, these resources have limitations that reduce the benefits obtained with their usage. One of these limitations is the resources' opening limited capacity that does not allow the own teachers to adapt them to their pedagogical needs. This adaptation idea is closely related to the vision of sharing and democratizing knowledge and information which the Open Educational Resources (OER) movement stands for. However, in order to make the adaptations according to this movement's philosophy, the current opening strategies require that the teachers access the resources source codes and have enough knowledge to change them. Another OA problem is the developing processes used that do not consider the multidisciplinary of the team involved in their production. Thus, this work applies software engineering concepts to reduce these limitations and proposes an adaptation strategy called Guided Customization. This strategy aims to indicate, in the design phase, the elements that can be handled by the teachers straightly in the resource's interfaces. A model-driven software development process with a domain specific language (DSL) is used to allow that and to improve the team communication. In order to implement the proposed strategy, an authoring tool (called CLO Studio) is developed and used to evaluate both the development of a Customizable Learning Object (CLO) by a LO development team and its usage by teachers. The results show that the CLO development process not only encourages the cooperation among the development team but also makes communication among them easier. Besides that, the proposed strategy gives new opportunities to the teachers' daily practices and improves their autonomy.

**Keywords:** Software Adaptation, Model-Driven Software Engineering, Learning Objects (LO), Development Process, Open Educational Resources (OER).

## LISTA DE FIGURAS

Figura 2.1 - Fases que compõem o modelo ADDIE. . . . .	34
Figura 2.2 - Modelo de Processo Labvirt . . . . .	36
Figura 2.3 - Modelo de Processo Rived . . . . .	37
Figura 2.4 - Modelo de Processo Rived Aprimorado . . . . .	38
Figura 2.5 - Modelo de Processo Colaborativo . . . . .	39
Figura 2.6 - Modelo de Desenvolvimento de um OA baseado na Aprendizagem Significativa . . . . .	40
Figura 2.7 - Etapas do Processo Sophia . . . . .	41
Figura 2.8 - Projeto de OA no Processo Sophia . . . . .	41
Figura 2.9 - Desenvolvimento de OA no Processo Sophia . . . . .	42
Figura 2.10 - Distribuição de OA no Processo Sophia . . . . .	43
Figura 2.11 - Modelo de Desenvolvimento MIDOA . . . . .	44
Figura 2.12 - Modelo de Desenvolvimento da UAA . . . . .	45
Figura 2.13 - Metodologia AODDEI . . . . .	46
Figura 2.14 - Relação de Dependência entre Internacionalização e Localização de Software. . . . .	54
Figura 2.15 - Tela Principal da Ferramenta de Autoria MLOAT . . . . .	57
Figura 2.16 - RED criado com a Ferramenta de Autoria MLOAT . . . . .	58
Figura 2.17 - Tela Principal da Ferramenta de Autoria GLO Maker . . . . .	59
Figura 2.18 - Tela com visão parcial do LOC mostrando orientação pedagógica para o desenvolvedor . . . . .	60

Figura 2.19 - Interface da Ferramenta RELOAD . . . . .	62
Figura 3.1 - Customização Guiada no Contexto de Desenvolvimento de um RED . . . . .	70
Figura 3.2 - Pontencial Pedagógico no SumaMonedas Vedoque . . . . .	75
Figura 3.3 - RED que Aborda a Compreensão e a aplicabilidade da Função Qua- drática . . . . .	76
Figura 3.4 - RED Fazenda Rived . . . . .	77
Figura 4.1 - Tela de Login do OAC Histórias Fantásticas . . . . .	88
Figura 4.2 - OAC Histórias Fantásticas com Grau de Liberdade 1 . . . . .	88
Figura 4.3 - OAC Histórias Fantásticas com Grau de Liberdade 3 . . . . .	89
Figura 4.4 - Processo Genérico de Produção de um OA Monolítico . . . . .	97
Figura 4.5 - Processo de Produção de um OAC . . . . .	98
Figura 4.6 - Cenas do OA Pontos em Batalha . . . . .	101
Figura 4.7 - Cena Interna do OA Pontos em Batalha . . . . .	101
Figura 4.8 - Representação da Arquitetura de uma Cena . . . . .	102
Figura 4.9 - Representação de uma Transição . . . . .	102
Figura 4.10 - Fragmento de um Roteiro de Atividades . . . . .	104
Figura 5.1 - Necessidades de uma DSL. . . . .	111
Figura 5.2 - Metodologia de Desenvolvimento do CLO Studio . . . . .	112
Figura 5.3 - Metamodelo ECore do CLO Studio . . . . .	122
Figura 5.4 - Interface do CLO Studio . . . . .	131
Figura 5.5 - Especificação do OAC do Exemplo 1 . . . . .	133
Figura 5.6 - Cena inicial da Simulação do Exemplo 1 . . . . .	134

Figura 5.7 - Cena resultante da ação de clicar do Exemplo 1 . . . . .	134
Figura 5.8 - Cena inicial do OAC do Exemplo 1 em execução . . . . .	135
Figura 5.9 - Cena resultante da ação de clicar no OAC do Exemplo 1 em execução	136
Figura 5.10 - Especificação do OAC do Exemplo 2 . . . . .	136
Figura 5.11 - Definição de Ação “Mouse Action” no CLO <i>Studio</i> . . . . .	137
Figura 5.12 - Cena inicial da Simulação do Exemplo 2 . . . . .	137
Figura 5.13 - Cena resultante da ação no botão “at1” do Exemplo 2 . . . . .	138
Figura 5.14 - Cena resultante da ação no botão “mostrar” do Exemplo 2 . . . . .	138
Figura 5.15 - Tela de um OAC em Execução . . . . .	140
Figura 5.16 - Barra de Customização de um OAC . . . . .	141
Figura 5.17 - Tela de Login em um OAC . . . . .	141
Figura 5.18 - Barra de Ações de Customização de um OAC. . . . .	142
Figura 5.19 - Conector da Barra de Ações de Customização de um OAC . . . . .	142
Figura 5.20 - Projeto de uma Cena com Vários tipos de Componentes . . . . .	149
Figura 5.21 - Simulação de uma Cena com Vários tipos de Componentes . . . . .	149
Figura 5.22 - Arquivo de Versões Gerado pelo do CLO Studio . . . . .	151
Figura 5.23 - Fragmento de um <i>Storyboard</i> Gerado pelo CLO <i>Studio</i> . . . . .	153
Figura 5.24 - Exemplo de Arquivo de Metadados LOM gerado pelo CLO Studio	154
Figura 5.25 - <i>Framework</i> de Suporte no Contexto de uma DSL . . . . .	155
Figura 5.26 - Janela de Seleção de Versão de OAC . . . . .	157
Figura 5.27 - Janela de Salvamento da Versão de OAC . . . . .	157
Figura 5.28 - Janela de Edição do Componente Texto de um OAC . . . . .	158
Figura 5.29 - Janela de Fragmentação de Vídeos de um OAC . . . . .	159

Figura 5.30 - Ferramenta de Reposicionamento e Redimensionamento do CLO <i>Studio</i> . . . . .	159
Figura 5.31 - Ferramenta de Conector de Componentes do CLO <i>Studio</i> . . . . .	160
Figura 6.1 - Projeto do OAC Pizza_Fração no CLO Studio . . . . .	185
Figura 6.2 - Simulação do OAC Pizza_Fração no CLO Studio . . . . .	186
Figura 6.3 - OAC Pizza_Fração em Execução . . . . .	186
Figura 6.4 - Tela Atividade 1 do OAC Histórias Fantásticas . . . . .	193
Figura 6.5 - Tela Atividade 1 do OAC Histórias Fantásticas (Customização 1) .	194
Figura 6.6 - Tela Atividade 1 do OAC Histórias Fantásticas (Customização 2) .	195
Figura 6.7 - Tela do Cenário FAZENDA do OAC Histórias Fantásticas . . . . .	196
Figura 6.8 - Tela Customizada do Cenário FAZENDA do OAC Histórias Fan- tásticas . . . . .	197
Figura 6.9 - Tela do Cenário FAZENDA com o conteúdo do botão AJUDA . .	198
Figura 6.10 - Tela Customizada do Cenário FAZENDA com o conteúdo do botão AJUDA . . . . .	199

## LISTA DE QUADROS

Quadro 2.1 - Análise Comparativa entre os Processos de Desenvolvimento de RED	51
Quadro 2.2 - Notação Matemática em Diferentes Países . . . . .	55
Quadro 2.3 - Análise Comparativa das Ferramentas de Autoria de RED . . . . .	64
Quadro 4.1 - Tipos de Elementos e Intervenções Adaptativas em um OAC . . . . .	85
Quadro 4.2 - Ações de Customização x Grau de Liberdade . . . . .	87
Quadro 4.3 - Comparação OA x OAC . . . . .	108
Quadro 5.1 - Ferramentas Utilizadas no Desenvolvimento do CLO Studio . . . . .	117
Quadro 5.2 - Notações do CLO Studio . . . . .	125
Quadro 5.3 - Metadados LOM dos OAC . . . . .	130
Quadro 5.4 - Elementos da Barra de Ações de Customização . . . . .	143
Quadro 5.5 - Elementos do Metamodelo do CLO <i>Studio</i> x Componentes da Linguagem <i>Flex</i> . . . . .	144
Quadro 6.1 - Materiais e Instrumentos Utilizados na Avaliação Empírica Qualitativa . . . . .	170
Quadro 6.2 - Estrutura do Grupo Focal de Avaliação da Concepção e Desenvolvimento . . . . .	175
Quadro 6.3 - Estrutura do Mini Grupo Focal de Avaliação da Utilização . . . . .	180
Quadro 6.4 - Contexto e Situações para a Avaliação na Utilização . . . . .	181
Quadro 6.5 - Avaliação das Hipóteses . . . . .	202

## LISTA DE ALGORITMOS

Algoritmo 5.1 - Algoritmo de Geração de <i>Templates</i> de Implementação do CLO <i>Studio</i> . . . . .	148
Algoritmo 5.2 - Algoritmo de Geração de <i>Storyboards</i> do CLO <i>Studio</i> . . . . .	152

# LISTA DE ACRÔNIMOS E SIGLAS

<b>ATL</b>	Atlas Transformation Language
<b>AVA</b>	Ambiente Virtual de Aprendizagem
<b>BIOE</b>	Banco Internacional de Objetos Educacionais
<b>CLAssRoOM</b>	Customizable LeARning Objects Model-driven
<b>CIM</b>	Computacional Independent Model
<b>DART</b>	Digital Anthropological Resources for Teaching
<b>DSL</b>	Domain Specific Language
<b>EMF</b>	Eclipse Modeling Framework
<b>EOL</b>	Epsilon Object Language
<b>EVL</b>	Epsilon Validation Language
<b>FOP</b>	Formatting Objects Process
<b>GD</b>	General Design
<b>GL</b>	Grau de Liberdade
<b>GLO Maker</b>	Generative Learning Object Maker authoring tool
<b>GEF</b>	Graphical Editing Framework
<b>GMF</b>	Graphical Modeling Framework
<b>GMP</b>	Graphical Modeling Project
<b>GMT</b>	Generative Modeling Tools
<b>GREat</b>	Grupo de Redes de Computadores Engenharia de Software e Sistemas
<b>ISD</b>	Instrucional Systems Design



<b>JET</b>	Java Emitter Templates
<b>Labvirt</b>	Laboratório Didático Virtual
<b>LOC</b>	Learning Object Creator
<b>LOCoMo</b>	Learning Objects Construction Methodology
<b>LOCPN</b>	Learning Objects production with Colored Petri Nets
<b>LOM</b>	Learning Object Metadata
<b>MDA</b>	Model-Driven Architecture
<b>MDD</b>	Model-Driven Development
<b>MDE</b>	Model-Driven Engineering
<b>MDSE</b>	Model-Driven Software Engineering
<b>Merlot</b>	Multimedia Educational Resource for Learning and Online Teaching
<b>MIDOA</b>	Modelo Instruiccional para el Desarrollo de Objetos de Aprendizaje
<b>MIT</b>	Massachusetts Institute of Technology
<b>MLOAT</b>	Multimedia Learning Object Authoring Tools
<b>MOF</b>	Meta-Object Facility
<b>OA</b>	Objetos de Aprendizagem
<b>OAC</b>	Objetos de Aprendizagem Customizáveis
<b>OC</b>	Open Content
<b>ODM</b>	Objetivos de Desenvolvimento do Milênio
<b>OER</b>	Open Educacional Resource
<b>OMG</b>	Object Management Group
<b>OSS</b>	Open Source Software

<b>PDF</b>	Portable Document Format
<b>PNG</b>	Portable Network Graphics
<b>PIM</b>	Platform Independent Model
<b>PNUD</b>	Programa das Nações Unidas para o Desenvolvimento
<b>PROATIVA</b>	Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem
<b>PSM</b>	Platform Specific Model
<b>QVT</b>	Queries/Views/Transformations
<b>REA</b>	Recurso Educacional Aberto
<b>REAC</b>	Recursos Educacionais com Adaptações Culturais
<b>RELOAD</b>	Reusable eLearning Object Authoring and Delivery
<b>RED</b>	Recursos Educacionais Digitais
<b>RESC</b>	Recursos Educacionais com Saliências Culturais
<b>SEED</b>	Secretaria de Educação a Distância
<b>Rived</b>	Rede Interativa Virtual de Educação
<b>TCS</b>	Textual Concrete Syntax
<b>UAA</b>	Universidad Autónoma de Aguascalientes
<b>UML</b>	Unified Modeling Language
<b>UNESCO</b>	Organização para a Educação, Ciências e Cultura
<b>XMI</b>	XML Metadata Interchange
<b>XML</b>	Extensible Markup Language

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>21</b>
1.1	Contexto . . . . .	21
1.2	Motivação . . . . .	25
1.3	Problemática . . . . .	25
1.4	Hipóteses . . . . .	27
1.5	Objetivos . . . . .	28
1.6	Contribuição e Relevância . . . . .	29
1.7	Método de Desenvolvimento da Tese . . . . .	30
1.8	Organização do Trabalho . . . . .	31
<b>2</b>	<b>Desenvolvimento e Adaptação de Recursos Educacionais Digitais</b>	<b>32</b>
2.1	Recursos Educacionais Digitais . . . . .	32
2.1.1	Processos de Desenvolvimento . . . . .	33
2.1.1.1	ADDIE . . . . .	34
2.1.1.2	LABVIRT . . . . .	35
2.1.1.3	RIVED . . . . .	36
2.1.1.4	PROATIVA . . . . .	38
2.1.1.5	Modelo de Processo Colaborativo . . . . .	38
2.1.1.6	Modelo de Desenvolvimento Baseado na Aprendizagem Significativa . . . . .	39
2.1.1.7	SOPHIA . . . . .	41

2.1.1.8	DART . . . . .	43
2.1.1.9	MIDOA . . . . .	43
2.1.1.10	Metodologia de Desenvolvimento da UAA . . . . .	45
2.1.1.11	OADDEI . . . . .	46
2.1.1.12	LOCoMo . . . . .	48
2.1.2	Análise Comparativa dos Processos de Desenvolvimento . . . . .	49
2.2	Adaptação de Recursos Educacionais Digitais . . . . .	52
2.2.1	Técnicas de Adaptação . . . . .	53
2.2.2	Ferramentas de Autoria . . . . .	56
2.2.2.1	<i>Multimedia Learning Object</i> (MLOAT) . . . . .	57
2.2.2.2	<i>Generative Learning Object Maker</i> (GLO Maker) . . . . .	58
2.2.2.3	<i>Learning Object Creator</i> (LOC) . . . . .	60
2.2.2.4	<i>Reusable eLearning Object Authoring &amp; Delivery</i> (RE- LOAD) . . . . .	61
2.2.2.5	Outras Ferramentas . . . . .	62
2.2.3	Análise Comparativa das Ferramentas de Autoria . . . . .	63
2.3	Discussão . . . . .	65
2.4	Conclusão . . . . .	66
<b>3</b>	<b>Customização Guiada</b> . . . . .	<b>68</b>
3.1	Justificativas e Conceito . . . . .	68
3.1.1	Concepção . . . . .	71
3.1.2	Utilização . . . . .	72
3.2	A Relação entre RED Customizável e REA . . . . .	73
3.3	Potencial Pedagógico . . . . .	74

3.4	Grau de Liberdade . . . . .	78
3.5	Discussão . . . . .	80
3.6	Conclusão . . . . .	81
<b>4</b>	<b>CLAssRoOM no Desenvolvimento de Objetos de Aprendizagem Customizáveis</b>	<b>82</b>
4.1	Objetos de Aprendizagem Customizáveis . . . . .	82
4.1.1	Definição de OAC . . . . .	83
4.1.2	Por que customizar . . . . .	84
4.1.3	O que customizar . . . . .	85
4.1.4	Vantagens e Limitações . . . . .	90
4.2	Como Construir Objetos de Aprendizagem Customizáveis . . . . .	91
4.2.1	Engenharia de Software Orientada a Modelos . . . . .	92
4.2.2	Processo CLAssRoOM . . . . .	96
4.2.2.1	Modelo de Arquitetura . . . . .	100
4.2.2.2	Arquitetura de Interface de Cenas . . . . .	102
4.2.2.3	Roteiro de Atividades . . . . .	103
4.2.2.4	<i>Templates</i> de Implementação . . . . .	105
4.2.2.5	Regras de Transformações . . . . .	105
4.3	Discussão . . . . .	106
4.4	Conclusão . . . . .	107
<b>5</b>	<b>CLO Studio: Ferramenta de Autoria para Produção de OAC</b>	<b>110</b>
5.1	Desenvolvimento do CLO Studio . . . . .	110
5.1.1	Metodologia . . . . .	112

5.2	Descrição dos Requisitos . . . . .	114
5.2.1	Objetivos . . . . .	114
5.2.2	Plataforma Alvo . . . . .	115
5.2.3	Ferramentas Utilizadas . . . . .	117
5.3	Projeto da Linguagem . . . . .	119
5.3.1	Conceitos Modelados . . . . .	120
5.3.2	Regras de Validação . . . . .	123
5.3.2.1	Regras Gerais . . . . .	123
5.3.2.2	Regras de Elementos . . . . .	124
5.3.2.3	Regras de Navegação . . . . .	124
5.3.3	Notação . . . . .	125
5.3.4	Interface do CLO Studio . . . . .	131
5.3.5	Modelos Exemplos . . . . .	133
5.3.5.1	Exemplo 1 . . . . .	133
5.3.5.2	Exemplo 2 . . . . .	135
5.4	Transformações . . . . .	139
5.4.1	Geração dos Templates de Implementação . . . . .	139
5.4.2	Geração de Arquivos de Simulação . . . . .	148
5.4.3	Geração do Arquivo de Versões . . . . .	150
5.4.4	Geração do Roteiro de Atividade ( <i>Storyboard</i> ) . . . . .	151
5.4.5	Geração de Metadados LOM . . . . .	153
5.5	<i>Framework</i> de Suporte . . . . .	155
5.5.1	Duplicação de Códigos Gerados . . . . .	156
5.5.2	Funcionalidades Adicionais . . . . .	158

5.6	Discussão . . . . .	160
5.7	Conclusão . . . . .	161
<b>6</b>	<b>Avaliação Empírica Qualitativa da Produção e Utilização de OAC</b>	<b>163</b>
6.1	Método de Avaliação . . . . .	164
6.1.1	Avaliação Empírica Qualitativa . . . . .	165
6.1.1.1	Métodos de Coleta de Dados . . . . .	167
6.1.1.2	Métodos de Análise de Dados . . . . .	169
6.1.2	Método Adotado . . . . .	169
6.2	Planejamento . . . . .	170
6.2.1	Materiais e Instrumentos Utilizados . . . . .	170
6.2.2	Planejamento da Avaliação na Concepção e Desenvolvimento . . . . .	172
6.2.2.1	Motivação e Hipóteses de Pesquisa . . . . .	173
6.2.2.2	Planejamento da Avaliação . . . . .	174
6.2.3	Planejamento da Avaliação na Utilização . . . . .	177
6.2.3.1	Motivação e Hipóteses de Pesquisa . . . . .	178
6.2.3.2	Planejamento da Avaliação . . . . .	179
6.3	Execução e Análise . . . . .	182
6.3.1	Execução da Avaliação na Concepção e Desenvolvimento . . . . .	182
6.3.1.1	OAC Produzido . . . . .	184
6.3.1.2	Resultados Alcançados . . . . .	187
6.3.2	Execução da Avaliação na Utilização . . . . .	191
6.3.2.1	OAC Utilizado . . . . .	191
6.3.2.2	Resultados Alcançados . . . . .	197
6.4	Discussão . . . . .	200

6.5	Conclusão . . . . .	201
<b>7</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>204</b>
7.1	Conclusão . . . . .	204
7.2	Trabalhos Futuros . . . . .	207
	<b>REFERÊNCIAS</b>	<b>211</b>
<b>A</b>	<b>Mematodelo Emfatic/Eugenia do CLO Studio</b>	<b>222</b>
<b>B</b>	<b>Regras de Validação em EVL</b>	<b>225</b>
<b>C</b>	<b>Scripts de Transformação</b>	<b>227</b>
C.1	Geração de Templates de Implementação-CLO2Flex.m2t . . . . .	227
C.2	Geração de Arquivos de Simulação-CLO2FlexSim.m2t . . . . .	234
C.3	Geração do Roteiro de Versões-CLO2Versions.m2t . . . . .	237
C.4	Geração do Roteiro de Atividades-CLO2Fop.m2t . . . . .	239
C.5	Geração de Metadados LOM-CLO2LOM.m2t . . . . .	241
<b>D</b>	<b>Roteiros de Entrevista</b>	<b>243</b>
D.1	Roteiro de Entrevista da Equipe de Desenvolvimento . . . . .	243
D.2	Roteiro de Entrevista do Professor . . . . .	247
<b>E</b>	<b>Diários de Campo</b>	<b>250</b>
E.1	Diário de Campo para a Equipe de Desenvolvimento . . . . .	250
E.2	Diário de Campo para o Professor . . . . .	251
<b>F</b>	<b>Roteiro de Atividades da Validação com o Professor</b>	<b>252</b>



# Capítulo 1

## Introdução

“O importante é não parar de questionar.”

– *Albert Einstein.*

Neste capítulo é apresentado o contexto ao qual esta pesquisa está inserida e os fatores que motivaram o desenvolvimento da mesma, mostrando com isso a problemática levantada e a hipótese tratada neste trabalho. Além disso, também são apresentados os objetivos a serem alcançados, a contribuição e a relevância desta pesquisa, tanto para o âmbito da engenharia de software quanto para o âmbito educacional, a estratégia adotada para o desenvolvimento da tese bem como o modo de organização do trabalho.

### 1.1 Contexto

A Educação pode ser considerada a chave para o desenvolvimento dos países (HATAKKA, 2009). No entanto, segundo relatório da Organização para a Educação, Ciências e Cultura (UNESCO), a sua efetivação ainda é um grave problema enfrentado por diversos países no mundo, que precisa ser solucionado para que os mesmos possam, de fato, se desenvolver (ALAM; ISMAIL; MISHRA, 2010; HALLAK, 1999). Prova disso, é que em pleno século 21 dentre os Objetivos de Desenvolvimento do Milênio (ODM), articulado pelo Programa das Nações Unidas para o Desenvolvimento (PNUD)<sup>1</sup>, a erradicação do analfabetismo ainda é tratado como uma das metas a ser atingida até 2015 pelos 191 estados membros das Nações Unidas. Essa meta deve ser viabilizada pelos países, através da criação de condições, para que todas as crianças possam obter o ensino básico. Contudo, a consecução dessa meta em países em desenvolvimento como

---

<sup>1</sup><http://www.pnud.org.br/odm/>

o Brasil, está ligada a diversos fatores, tais como as taxas de frequência escolar e de tempo médio de conclusão do ensino fundamental.

Dentre as diversas estratégias que podem ser usadas para apoiar a viabilização desse objetivo, é importante que os educadores utilizem recursos que motivem os alunos e os mantenham interessados ao longo de todo o período em que permanecem nas escolas. Neste sentido, diversos trabalhos indicam o uso de tecnologias como ferramental que apresenta resultados efetivos nesta direção (FERNANDES; COSTA, 2009; BAKI; ÇAKIROGLU, 2010). Entre as alternativas tecnológicas utilizadas atualmente no contexto da sala de aula, os Recursos Educacionais Digitais (RED), tais como imagens, vídeos, hipertextos e softwares, têm sido amplamente valorizados, principalmente pelo fato de que esses recursos podem ser facilmente disponibilizados e acessados através da Internet. Assim, além de ser uma fonte de motivação para professores e alunos, os RED também se apresentam como ferramentas efetivas de aprendizagem (BAKI; ÇAKIROGLU, 2010).

Um tipo de RED amplamente difundido na Internet são os Objetos de Aprendizagem (OA) (LONGMIRE, 2000). Esses recursos são ferramentas para auxiliar o professor a criar novas estratégias de ensino que favoreçam a abstração do conhecimento pelo aluno.

Para Shepherd (2000) e Willey (2001), os OA são aplicações resultantes da orientação a objetos no mundo da aprendizagem e são pequenos componentes reusáveis (vídeo, demonstrações, tutoriais, procedimentos, histórias e simulações) que não servem simplesmente para produzir ambientes, e sim, para desenvolver pessoas. Esse desenvolvimento ocorre porque os OA fornecem cenários interessantes e com isso são capazes de ajudar na compreensão dos problemas e conceitos (BAKI; ÇAKIROGLU, 2010).

Essa valorização dos RED de um modo geral, e, particularmente dos OA, é decorrente da possibilidade de trazer as experiências do mundo real para sala de aula através da tecnologia, proporcionando assim um andaime que permite aos alunos participarem de tarefas cognitivas complexas, aumentando as oportunidades para receberem sofisticados e individualizados *feedbacks*, além de construir comunidades de interação entre professores, alunos, pais e outros grupos interessados e de permitir a ampliação das oportunidades para o desenvolvimento do professor (BAEK; JUNG; KIM, 2008 apud DONOVAN; BRANSFORD; PELLEGRINO, 2000).

Na literatura, os RED podem ser chamados de conteúdos pedagógicos digitais ou materiais educacionais digitais. Independente da denominação adotada, esses materiais, conteúdos ou recursos buscam, por intermédio da tecnologia, serem apresentados de modo a facilitar sua compreensão.

Diante dessas inúmeras vantagens, os órgãos governamentais, universidades e instituições privadas vêm ampliando e incentivando as condições de uso desses recursos em sala de aula como ferramenta auxiliar ao processo de ensino através da criação de repositórios e cursos. No entanto, uma das grandes preocupações relacionadas a esses recursos, diz respeito à forma como estes são, uma vez desenvolvidos, encontrados e acessados.

Como forma de investir no acesso e, conseqüentemente, na disseminação de tecnologias educacionais, foram desenvolvidos espaços destinados ao armazenamento e busca de RED, denominados de repositórios. Os repositórios existem tanto no âmbito nacional quanto internacional. No âmbito nacional é possível citar o projeto Rede Interativa Virtual de Educação (Rived) como um dos grandes investimentos do governo brasileiro. Este projeto, desenvolvido inicialmente em parceria com o Peru e a Venezuela no ano de 1999, teve por objetivo a produção e disponibilização pública de OA. Com a saída dos países parceiros ainda no início do projeto, o governo brasileiro resolveu financiar sozinho a criação dos recursos através de editais endereçados às Universidades Públicas Brasileiras.

Posteriormente, no ano de 2008, o governo brasileiro deu início a um novo projeto, em substituição ao Rived. Com o propósito de buscar e catalogar conteúdos digitais de qualquer parte do mundo, o Banco Internacional de Objetos Educacionais (BIOE)<sup>2</sup> foi criado como um repositório de acesso livre alimentado pelas universidades públicas que através de financiamento governamental realizam buscas e catalogam os recursos dos mais diversos formatos.

Com relação aos investimentos governamentais em nível internacional, é possível mencionar a criação da Federação da Aprendizagem (*Le@rning Federation*)<sup>3</sup>. A Le@arning Federation é resultado de investimentos de fundos do governo da Austrália e Nova Zelândia e tem por objetivo desenvolver colaborativamente uma fonte contínua de recursos digitais de alta qualidade, através de uma série de projetos, e fornecê-los às escolas australianas e neozelandesas.

No âmbito das universidades, é possível citar o MIT *Open Courseware*<sup>4</sup>, também conhecido como MIT OCW. Esta iniciativa é uma ação conjunta do MIT com as funda-

---

<sup>2</sup><http://objetoseducacionais2.mec.gov.br>

<sup>3</sup><http://www.thelearningfederation.edu.au>

<sup>4</sup><http://ocw.mit.edu/>

ções William and Flora Hewlett e Andrew W. Mellon e tem por objetivo proporcionar acesso gratuito aos materiais relativos aos cursos do MIT para educadores de instituições sem fins lucrativos, estudantes e autodidatas do mundo todo. Além de criar um modelo padronizado para que outras universidades possam copiar para publicar os materiais de seus próprios cursos.

Ainda no âmbito internacional, mas agora na esfera das instituições privadas, é possível citar o *iCampus Project*<sup>5</sup>. O iCampus é um projeto desenvolvido pela Microsoft Research em conjunto com o *Massachusetts Institute of Technology* (MIT) e tem por objetivo expandir o acesso aos cursos desta instituição através da disponibilização de laboratórios virtuais e outras tecnologias desenvolvidas no MIT.

Além dos já citados, existem diversos repositórios espalhados pelo mundo. No Reino Unido é possível citar o Jorum<sup>6</sup>. Na Califórnia é possível citar o *Multimedia Educational Resource for Learning and Online Teaching* (Merlot)<sup>7</sup> dentre outros.

Devido a essa grande disponibilidade de recursos inseridos nos mais diversos repositórios existentes, surge agora a preocupação de consolidar não apenas o compartilhamento dos recursos, mas também o compartilhamento de conhecimentos. Diante dessa necessidade, surge o movimento em defesa da abertura dos RED, denominado *Open Educational Resources* (OER). Esse movimento compartilha dos ideais do *Open Source*, e defende que os materiais educacionais digitais devam ser desenvolvidos de modo aberto e disponibilizados de forma livre, para que possam ser utilizados pela comunidade acadêmica em geral, no ensino, aprendizagem e pesquisa.

Segundo Hylén (2006), o OER é um fenômeno relativamente novo, que pode ser visto como parte de uma tendência maior para a abertura em ensino superior, incluindo movimentos mais conhecidos mundialmente e estabelecidos, como *Open Source Software* (OSS) e o *Open Access* (OA). A abertura proposta por esse movimento foca na liberdade, herdada do movimento do software livre, para alterar o código fonte do recurso.

Nesse contexto, professores e alunos se deparam atualmente com conceitos de conteúdo aberto, recursos abertos, ensino aberto, arquitetura aberta no design de espaços físicos e virtuais, código aberto, e conhecimento aberto (BATSON; PAHARIA; KUMAR, 2008). Contudo, é preciso ampliar a discussão sobre o uso dessas tecnologias visando compreender as limitações do tipo de abertura proposto ao ensino e como os recursos educacionais são projetados, desenvolvidos e utilizados por aqueles que buscam por novas oportunidades de ensino e aprendizagem.

---

<sup>5</sup><http://icampus.mit.edu/>

<sup>6</sup><http://www.jorum.ac.uk/>

<sup>7</sup><http://www.merlot.org/>

## 1.2 Motivação

Segundo Baek, Jung e Kim (2008), os professores têm interesse em utilizar os RED em sala de aula. No entanto, como apontado por alguns pesquisadores, há muitos obstáculos que os impedem de utilizá-los, como por exemplo, infraestrutura inadequada (MEHLINGER; POWERS, 2002; PELGRUM, 2001), falta de formação e conhecimentos pessoais (JACOBSON; WELLER, 1988; SCHRUM, 1999; STRUDLER; WETZEL, 1999; WILLIS; THOMPSON; SADERA, 1999), e suporte técnico fraco (NETDAY, 2001; SCHRUM, 1995), dentre outros, são os desafios mais citados.

Para Almeida (2000) e Pelgrum (2001) um dos maiores obstáculos diz respeito à autonomia exercida pelos professores, pois a falta de formação e conhecimentos pessoais tanto no domínio do computador quanto no software, acaba por se constituir como a maior ansiedade por parte desses profissionais.

Almeida (2000) ressalta ainda que não basta simplesmente utilizar a tecnologia em sala de aula, é necessário observar que somente pelo caminho da democracia, da participação, da liberdade responsável e do desenvolvimento da autonomia será possível evitar os efeitos decepcionantes de sua incorporação no ensino e na aprendizagem.

Nesse contexto de abertura e colaboração em prol do compartilhamento de informações e conhecimento, o surgimento do OER proporcionou avanços na forma de pensar e desenvolver recursos educacionais digitais. Muito embora ainda haja limitações na autonomia proporcionada por esse movimento, tendo em vista que apenas uma parcela dos que utilizam esses recursos tem condições de usufruir dessa autonomia. Essa parcela está relacionada àqueles que têm conhecimento técnico para realizar modificações em nível de código, que é a forma pela qual essa abertura se dá para seus usuários.

No entanto, por se tratarem de recursos educacionais, as modificações a serem realizadas, muitas vezes, são bastante específicas e por isso requerem a intervenção direta do professor, que na maioria das vezes, não possui qualificação para realizar esse tipo de intervenção. Além disso, mesmo que essa qualificação exista, modificações realizadas de forma indiscriminada podem alterar completamente os objetivos pedagógicos dos recursos. Dessa forma, essas questões acabam por delimitar a problemática tratada nesta tese de doutorado e apresentada a seguir.

## 1.3 Problemática

Conforme mencionado em diversos trabalhos, existem diversas vantagens em se utili-

zar RED em sala de aula (INAN; LOWTHER, 2010; RUTTEN; JOOLINGEN; VEEN, 2011; ERTMER; OTTENBREIT-LEFTWICH, 2010; MEANS; PLANNING; DEVELOPMENT, 2009; PATRICK, 2008).

Contudo, os RED muitas vezes não são adequados para responder às necessidades específicas dos professores (LAWLESS; PELLEGRINO, 2007), pois situações como o excesso de atividades disponibilizadas no recurso, o sequenciamento das informações apresentadas, os conteúdos e a qualidade dos textos de *feedback*, dentre outras, podem influenciar na decisão sobre a adoção de um RED. Dessa forma, o professor tem a opção de adequar o seu planejamento ao RED disponível ou simplesmente desistir de utilizá-lo

Nem mesmo se o recurso for aberto, como pregado pelo movimento OER, o problema é resolvido, visto que o professor, na maioria das vezes, não tem conhecimento técnico para realizar as alterações necessárias para adaptá-lo às suas necessidades (MUELLER et al., 2008; PRESTRIDGE, 2011). Portanto a não adoção do RED é o mais provável.

Com base nessas constatações foi levantada a seguinte questão de pesquisa:

---

**Como produzir RED que possam ser ajustados diretamente pelos professores, levando em consideração o conhecimento técnico desses profissionais, sem, no entanto, comprometer as características pedagógicas associadas ao recurso na sua produção?**

---

Para que essa questão de pesquisa seja tratada, alguns pontos importantes devem ser esclarecidos. Inicialmente a produção de um RED envolve um conjunto de profissionais que trabalham de forma coordenada para desenvolver um produto que, além de ser um recurso tecnológico, carrega consigo um importante teor pedagógico. Essa característica é fundamental para que um RED seja efetivo na sua utilização. Portanto, a disponibilização de um RED que possa ser manipulado diretamente pelos professores deve ser feito de forma responsável pela equipe de produção. Desse modo, as ações passíveis de serem realizadas pelos professores devem ser devidamente analisadas por todos os profissionais envolvidos ainda em tempo de concepção. Portanto, o processo de desenvolvimento de um RED com essas características exige um incremento da integração e da comunicação entre os profissionais envolvidos.

Além da produção de RED passíveis de adaptação, os professores devem ser capazes de realizar essas adaptações de forma simples e direta, visto que a ampliação das

oportunidades de utilização de um RED está ligada à sua autonomia em realizar os ajustes necessários para que esses recursos se encaixem em seu planejamento.

Dessa forma, no intuito de responder à questão levantada, é proposto nesta tese a estratégia de customização guiada. A ideia de guiada é norteada por dois princípios. O primeiro diz respeito a guiar, os que não têm conhecimento técnico para modificar o código, na realização de modificações a serem feitas na interface do RED. E o segundo, diz respeito à preservação dos objetivos pedagógicos do recurso.

Para viabilizar a estratégia da customização guiada é utilizada uma abordagem baseada em Engenharia de Software Orientada a Modelos, também denominada *Model-Driven Software Engineering* (MDSE) (SCHMIDT, 2006), a ser aplicada no processo de produção dos RED, mais especificamente os OA do tipo animação/simulação. Essa abordagem faz uso de mecanismos de abstração, através de Linguagens Específicas de Domínio ou *Domain Specific Language* (DSL), que têm por finalidade suplantam as dificuldades subjacentes ao desenvolvimento desse tipo de recurso.

A escolha por MDSE foi motivada pelo fato de ser importante a criação de um mecanismo de comunicação que possa ser utilizado e compreendido por todos os envolvidos na produção dos recursos, visto que profissionais de diferentes formações, tais como pedagogos e especialistas de área, devem participar desse processo. Desse modo, por fornecer mecanismos de abstração através de linguagens de fácil manipulação, a MDSE permite a manipulação colaborativa dos conceitos relacionados aos recursos e às estratégias de customização necessárias à sua adaptação.

## 1.4 Hipóteses

Autonomia é uma palavra de origem grega e significa o poder de dar a si a própria lei, *autós* (por si mesmo) e *nomos* (lei). Como a autonomia é "condição", como ela se dá no mundo e não apenas na consciência dos sujeitos, sua construção envolve dois aspectos: o poder de determinar a própria lei e também o poder ou capacidade de realizar. O primeiro aspecto está ligado à liberdade e ao poder de conceber, fantasiar, imaginar, decidir, e o segundo ao poder ou capacidade de fazer. Para que haja autonomia os dois aspectos devem estar presentes, e o pensar autônomo precisa ser também fazer autônomo (ZATTI, 2007).

Como forma de tratar esse problema, esta tese propõe a utilização da customização guiada. A customização, mesmo sendo uma ação executada pelo professor, é oriunda

da concepção do recurso. Dessa forma, como apresentado na Seção 1.3, é necessário um incremento da comunicação entre os membros da equipe de produção do recurso para que esses aspectos possam ser devidamente planejados. Como no contexto deste trabalho o processo de produção utiliza a abordagem da engenharia de software orientada a modelos, a seguinte hipótese pode ser firmada:

---

**A Engenharia de Software Orientada a Modelos possibilita uma melhoria na comunicação entre os membros da equipe de produção, favorecendo um melhor tratamento dos aspectos de customização guiada nestes recursos.**

---

Ainda com relação a autonomia e dentro dessa perspectiva do pensar e fazer autônomo, é possível entender que a oferecida pelo movimento OER não contempla todo e qualquer professor, haja vista que a **liberdade** ofertada para modificar um recurso atinge um público restrito com conhecimento e **capacidade** para fazer as alterações.

No intuito de fornecer autonomia a todo e qualquer professor, que utilizam os RED, mais especificamente os OA do tipo animação/simulação, como ferramenta auxiliar na transmissão dos conceitos, pode ser firmada a segunda hipótese desta tese:

---

**A utilização de recursos adaptáveis (OA customizáveis) em sala aula eleva a autonomia dos professores na criação de novas oportunidades e estratégias de ensino.**

---

## 1.5 Objetivos

A pesquisa desta tese de doutorado tem como objetivo geral propor uma estratégia de adaptação, denominada Customização Guiada, a ser incorporada ao processo de desenvolvimento de RED do tipo OA, que dá condições de ampliar tanto a autonomia docente na criação de novas oportunidades de ensino, quanto à capacidade de abertura desses recursos. O processo, por sua vez, deve favorecer a comunicação entre os membros da equipe de produção. Como forma de direcionar as ações da pesquisa, o foco principal desse trabalho está em RED do tipo OA, que são denominados de Objetos de Aprendizagem Customizáveis (OAC). Desse modo, os seguintes objetivos específicos podem ser listados:



1. Propor uma estratégia de customização de RED na qual o professor possa de fato realizar adaptações nos recursos sem, no entanto, interferir na qualidade pedagógica dos mesmos;
2. Propor um processo de desenvolvimento de OAC que forneça suporte à produção de recursos customizáveis;
3. Implementar uma ferramenta para permitir a automatização do processo proposto para auxiliar o desenvolvimento de OAC por parte de uma equipe multidisciplinar.

## 1.6 Contribuição e Relevância

Com base nos objetivos definidos e na problemática levantada a respeito do nível de abertura oferecido pelo movimento OER para modificar um RED do tipo OA, e pelo fato do mesmo ser considerado um software, a presente tese de doutorado descreve a seguir, um resumo das contribuições a serem alcançadas com esse trabalho.

- **Proposta da Estratégia da Customização Guiada:** A relevância dessa contribuição está no fato de que a utilização desta estratégia possibilita a criação de OA do tipo animação/simulação que desde a sua concepção tem por preocupação o fornecimento de mecanismos que possibilitem os professores utilizarem de forma efetiva esses recursos em suas ações pedagógicas, possibilitando a apropriação do recurso por parte do professor, que não mais terá que adaptar suas aulas aos recursos disponíveis, mas terá a oportunidade de adaptar o recurso às suas aulas aumentando a sua autonomia.
- **Utilização de Engenharia de Software Orientada à Modelos na Criação de OA:** além de elevar o nível de abstração na criação dos recursos, a utilização de modelos possibilita a geração automática dos códigos dos recursos. Essa automatização permite a preservação das características pedagógicas dos mesmos descritas pela equipe de projeto. A relevância dessa ação acontece pelo fato de que normalmente existem ruídos no processo de desenvolvimento de qualquer software e, especificamente no caso de um OA que tem funções pedagógicas como objetivo principal, esses ruídos de comunicação podem ser extremamente prejudiciais à compreensão dos conceitos.

- **Desenvolvimento de uma Ferramenta para dar suporte à criação de OA customizáveis:** como o objetivo desta tese é produzir um recurso que seja adaptável de forma planejada pelos professores em sala de aula, o processo de desenvolvimento do recurso é de fundamental importância. Deste modo, uma ferramenta está sendo proposta neste trabalho para dar suporte a esse desenvolvimento. Essa ferramenta é baseada em uma Linguagem Visual de Domínio Específico que permite a realização do projeto de arquitetura dos recursos. Essa linguagem, além de possibilitar a descrição dos elementos formadores dos recursos, permite a identificação das ações de customização passíveis de serem realizadas pelos usuários em tempo de utilização dos recursos. Além disso, outros recursos foram incorporados a ferramenta proposta para dar apoio à criação dos materiais educacionais, tais como *scripts* de geração automática de código, *scripts* de geração de documentação dos recursos, *scripts* de validação das especificações, dentre outros. Adicionalmente, pelo fato de serem baseadas em um conjunto de elementos gráficos de alto nível, tais recursos auxiliam na comunicação entre os atores participantes no processo de desenvolvimento dos recursos.
- **Aplicação da Customização Guiada à OA do tipo Animação/Simulação:** os OA são RED bastante utilizados no contexto de sala de aula. Neste sentido, a utilização da estratégia da customização guiada é demonstrada nesta tese para esse tipo de recurso, cujo resultado final é um tipo específico de OA, denominado neste trabalho de Objetos de Aprendizagem Customizáveis (OAC).

## 1.7 Método de Desenvolvimento da Tese

O método adotado para o desenvolvimento desta tese foi organizado em oito passos, descritos a seguir:

1. Estudo sobre o desenvolvimento de OA;
2. Levantamento Bibliográfico sobre os processos de desenvolvimento e Adaptação de RED;
3. Estudo sobre Engenharia de Software Orientada a Modelos;
4. Proposta da Estratégia da Customização Guiada;
5. Aplicação da Customização Guiada no desenvolvimento de OA;

6. Proposta de um processo de desenvolvimento de OAC utilizando Engenharia de Software Orientada a Modelos;
7. Desenvolvimento de uma ferramenta de suporte ao desenvolvimento de OAC;
8. Realização de uma avaliação das contribuições utilizando uma estratégia empírica qualitativa;
9. Análise dos resultados obtidos com a aplicação do estudo de caso.

## 1.8 Organização do Trabalho

O restante desta tese está dividida da seguinte forma:

- No Capítulo 2 é apresentado à fundamentação teórica e os trabalhos relacionados ao desenvolvimento e adaptação de recursos educacionais digitais.
- No Capítulo 3 é descrita a estratégia de customização guiada de recursos educacionais digitais com foco em OA, que é a principal contribuição desta tese.
- No Capítulo 4 é descrito um processo de desenvolvimento orientado a modelos de objetos de aprendizagem para viabilizar a estratégia da customização guiada
- No Capítulo 5 é apresentada a implementação de uma ferramenta de autoria para a construção de objetos de aprendizagem customizáveis.
- No Capítulo 6 é feita a avaliação das contribuições desse trabalho, tanto no aspecto da ferramenta utilizada para produzir os recursos quanto na utilização dos conceitos propostos.
- Por fim, no Capítulo 7 são descritas as conclusões deste trabalho, bem como as perspectivas futuras.

## Capítulo 2

# Desenvolvimento e Adaptação de Recursos Educacionais Digitais

“You have your way. I have my way. As for the right way, the correct way, and the only way, it does not exist.”  
– *Friedrich Nietzsche*

Neste capítulo são apresentados a definição, a classificação e os processos de desenvolvimento empregados na produção dos Recursos Educacionais Digitais (RED), juntamente com técnicas de adaptação e ferramentas de autoria que dão suporte a criação desses recursos.

### 2.1 Recursos Educacionais Digitais

Conforme descrito no Capítulo 1 desta tese, existem diversos tipos RED e estes são utilizados por professores e educadores como ferramentas adicionais, empregadas em sala de aula, para auxiliar na transmissão e compreensão dos conceitos.

Esta tese centra sua discussão em um tipo específico denominado de Objetos de Aprendizagem (OA) do tipo animação/simulação, visto que a autora tem experiência com a produção e a utilização desses recursos e pelo fato da literatura tratá-los como recursos capazes de reproduzir em sala de aula as experiências do mundo real (contextualização e interação). Essa capacidade de interação e principalmente de contextualização, acabam suscitando a necessidade de abertura para modificar o recurso. Além, da capacidade de reprodução, os OA tem a reutilização como um de seus potenciais (LONGMIRE, 2000) e para que isto aconteça de fato, é importante que em seu desenvolvimento não haja falhas na compreensão resultando em uma correta interpre-

tação e consequente produção do que foi especificado.

A necessidade de ampliar a capacidade de adaptação dos recursos educacionais e a consequente reutilização dos OA acabou motivando a filosofia de desenvolvimento dos mesmos, denominada de Recursos Educacionais Abertos (REA) (ALBRIGHT, 2005).

Segundo Hylén (2006), os REA surgiram pela primeira vez em 2002 durante uma conferência da UNESCO que os definiu como materiais educacionais digitais disponibilizados de forma livre e aberta para a comunidade acadêmica em geral, que os utilizam para o ensino, aprendizagem e pesquisa.

Os REA, conhecidos internacionalmente como *Open Educational Resources* (OER), acompanham as filosofias e movimentos a favor da abertura do conhecimento (*Open Source Software* (OSS), *Open Access* (OA), *Open Content* (OC), entre outros) (HYLÉN, 2006).

De acordo com a definição apresentada pela UNESCO, os OA fazem parte da categoria dos REA. No entanto, eles precisam ser pensados não somente como recursos que podem ser re combinados, mas também reutilizados em contextos diferentes, permitindo também que modificações sejam feitas por outros usuários (LONGMIRE, 2000) (ALBRIGHT, 2005).

Dentro dessa preocupação, é discutido na seção 2.1.1 os processos empregados e suas limitações no desenvolvimento de OA, além do modo como esses recursos são pensados e projetados para contemplar as necessidades daqueles que fazem uso desse ferramental.

### 2.1.1 Processos de Desenvolvimento

Na literatura existem diversos processos de desenvolvimento que são voltados para a produção de RED. Esses recursos são materiais que, em sua maioria, são desenvolvidos sob o formato de texto, vídeo, áudio ou ainda no formato de objetos de aprendizagem (OA). Com base nessa diversidade de recursos, são apresentados neste capítulo, alguns dos processos de desenvolvimento evidenciados na literatura empregados para a produção dos mesmos.

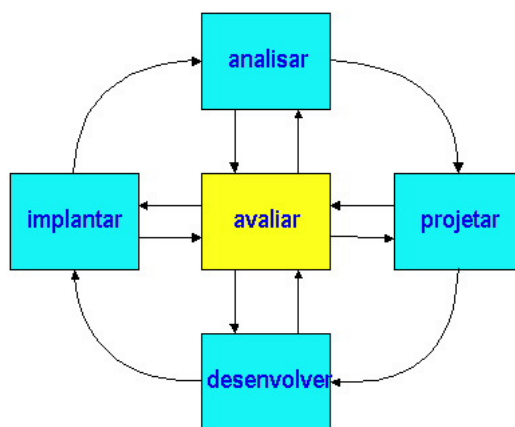
### 2.1.1.1 ADDIE

O ADDIE é um processo desenvolvido com base no *Instrucional Systems Design* (ISD) e tem por objetivo planejar, projetar e avaliar a eficácia do material instrucional.

O ISD é uma metodologia de desenvolvimento de processos instrucionais que tem seu desenvolvimento realizado, basicamente, em três etapas inter-relacionadas. São elas: conhecimento do público-alvo (identificação das necessidades), proposta de solução para estas necessidades (desenho da solução) e avaliação dos resultados (WALLL; MARCUSSO; TELLES, 2006).

Com base nas etapas inter-relacionadas do ISD, o modelo ADDIE é composto por 5 etapas: análise, design, desenvolvimento, implementação e avaliação. Essas etapas, como apresentado na Figura 2.1, são dependentes entre si, pois cada uma alimenta a seguinte, sendo que caso a etapa anterior não esteja definitivamente concluída, as outras ficam seriamente comprometidas.

Figura 2.1: Fases que compõem o modelo ADDIE.



Fonte: (WALLL; MARCUSSO; TELLES, 2006)

Para Pessoa e Benitti (2008), Mustaro et al. (2007), a utilização do modelo ADDIE pode também ser empregada no desenvolvimento de Objetos de Aprendizagem. A seguir, são apresentados os cinco estágios do modelo ADDIE sob a percepção de Mustaro et al. (2007):

**Análise** (*Analysis*): estágio caracterizado basicamente pela determinação e análise dos objetivos (principalmente educacionais) do Objeto de Aprendizagem. Envolve

geralmente pré-requisitos, análise do reuso do objeto, o cenário tecnológico, as mídias mais apropriadas, os envolvidos no projeto;

**Projeto** (*Design*): este estágio acontece paralelamente à fase de Análise, objetivando ajustar tópicos do projeto antes da produção do Objeto de Aprendizagem. Compreende basicamente os planos de construção do cenário do objeto. Também define estratégias, interatividade, feedback, usabilidade, entre outros;

**Desenvolvimento** (*Development*): estágio que há eventos relatados para a criação de objetos de aprendizagem. São alguns passos que podem ser executados: interface, produção de elementos de mídia, gerar ou conectar exercícios, desenvolvimento de testes de usabilidade, programação de interação entre objetos, navegação de formulários, *feedback* e retorno para o aluno;

**Implementação** (*Implementation*): esta fase necessita que um Objeto de Aprendizagem esteja em um ROA (Repositório de Objetos de Aprendizagem). Há três pontos que são relevantes nesta etapa: incluir metadados no objeto, testar o objeto em diversos ambientes e testar a interface do ROA para com o objeto;

**Avaliação** (*Evaluation*): esta é a última fase, que compreende todo o processo de desenvolvimento. Nela, o objeto de aprendizagem é avaliado em questão do aprendizado que pode proporcionar. Essa avaliação deve ocorrer com o auxílio de alunos e professores.

Diante do crescimento da produção de materiais instrucionais do tipo OA, vários foram os processos de desenvolvimento que surgiram, tanto no âmbito nacional quanto internacional, com o intuito de produzir materiais de qualidade. Com base nesse tipo de material, são descritos a seguir, outros processos de desenvolvimento utilizados na produção de OA.

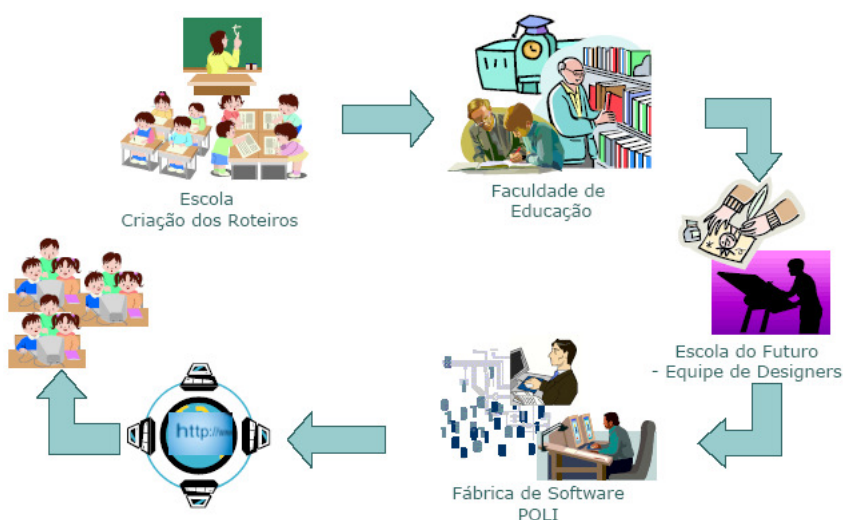
#### 2.1.1.2 LABVIRT

O Laboratório Didático Virtual (Labvirt) é uma iniciativa nacional realizada pela Microsoft do Brasil em parceria com a Escola do Futuro da Universidade de São Paulo

para o desenvolvimento e armazenamento de OA, desenvolvidos para o ensino médio, especificamente nas áreas de química e física <sup>1</sup>.

O modelo de desenvolvimento de OA adotado pelo Labvirt, apresentado na Figura 2.2, utiliza uma estrutura composta pelos seguintes atores: alunos e professores de escolas públicas, professores da universidade (núcleo), designers gráficos e programadores. Cada ator é responsável por compor uma equipe e essa, por sua vez, responsável por desenvolver uma atividade. Dessa forma, professores e alunos das escolas públicas consultam os professores do núcleo sobre a possibilidade de produção de um determinado conteúdo. Essa solicitação é entregue ao professor de núcleo, via *storyboard* que, por sua vez, trata de analisar a viabilidade no atendimento da solicitação. Confirmada a viabilidade, o *storyboard* é repassado ao designer gráfico para que contextualize o conteúdo a partir da criação dos cenários. Finalizada a etapa de design os programadores dão início à etapa de produção do objeto. Uma vez produzido, o recurso é disponibilizado na *web* para que os alunos e professores possam utilizá-lo.

Figura 2.2: Modelo de Processo Labvirt



Fonte: (BECERRA J. R.; SIQUEIRA, 2005)

### 2.1.1.3 RIVED

A Rede Interativa Virtual de Educação (RIVED) é um programa da Secretaria de Educação a Distância (SEED), que tem por objetivo a produção de conteúdos pedagó-

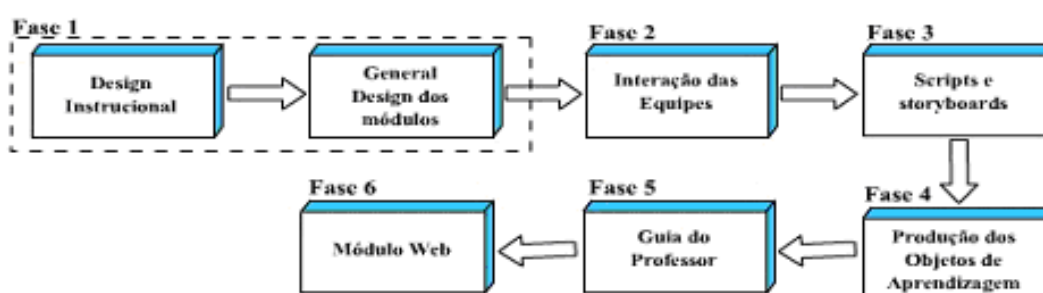
<sup>1</sup><http://www.labvirt.fe.usp.br>



gicos digitais, na forma de OA baseados em simulações e animações <sup>2</sup>. Tais conteúdos primam por estimular o raciocínio e o pensamento crítico dos estudantes, associando o potencial da informática às novas abordagens pedagógicas.

O modelo de processo proposto pelo RIVED, apresentado na Figura 2.3, é composto por seis fases e baseado na junção de um conjunto de características que são pertinentes ao aprendizado do indivíduo. Essa junção dá origem aos documentos *General Design (GD)* ou Design Pedagógico, Roteiro de Atividades e Guia do Professor.

Figura 2.3: Modelo de Processo Rived



Fonte: Adaptado de Nascimento e Morgado (2003)

Na Fase 1 desse modelo, os objetivos e as atividades são descritas no documento intitulado *General Design (GD)*. Na Fase 2, o GD é submetido às outras equipes, que participam do processo de desenvolvimento, para obter críticas e feedback. Na Fase 3, os especialistas de conteúdo revisam o design original após receber *feedback* das outras equipes. Nesse momento, eles (os especialistas de conteúdo) descrevem as especificações do OA para que a equipe técnica implemente o produto anteriormente especificado, dando origem ao documento Roteiro de Atividades. Na Fase 4, a equipe técnica implementa o OA. Na Fase 5, os especialistas em conteúdo criam o Guia do Professor do OA desenvolvido. Por fim, na Fase 6, os OA são organizados em módulos e publicados na web.

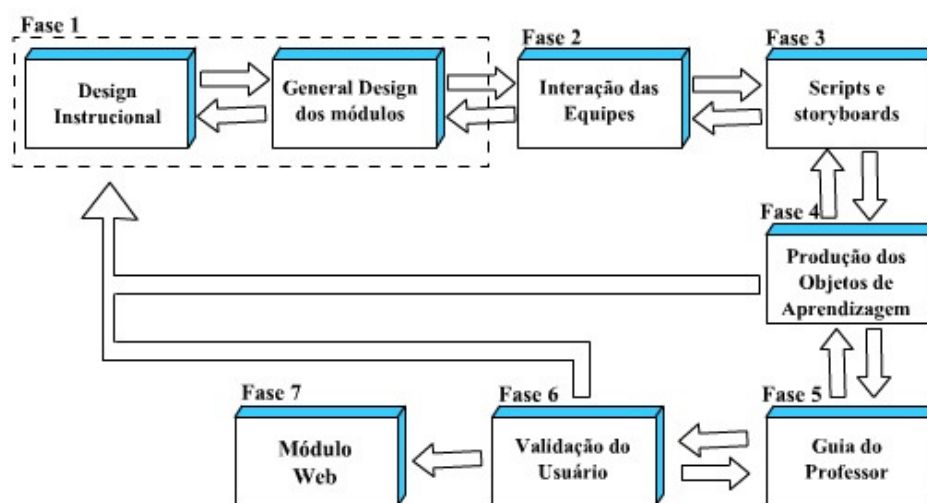
O processo RIVED serviu de base para criação de outros processos de desenvolvimento, como por exemplo, o processo Proativa e o Processo Colaborativo descritos a seguir.

<sup>2</sup><http://www.rived.mec.gov.br>

### 2.1.1.4 PROATIVA

Em Amaral et al. (2006) foi apresentado um aprimoramento do processo proposto pelo RIVED, em que há uma preocupação relevante na etapa de validação dos produtos desenvolvidos. Esse modelo, apresentado na Figura 2.4, é adotado, atualmente, na produção de OA do Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (PROATIVA)<sup>3</sup>.

Figura 2.4: Modelo de Processo Rived Aprimorado



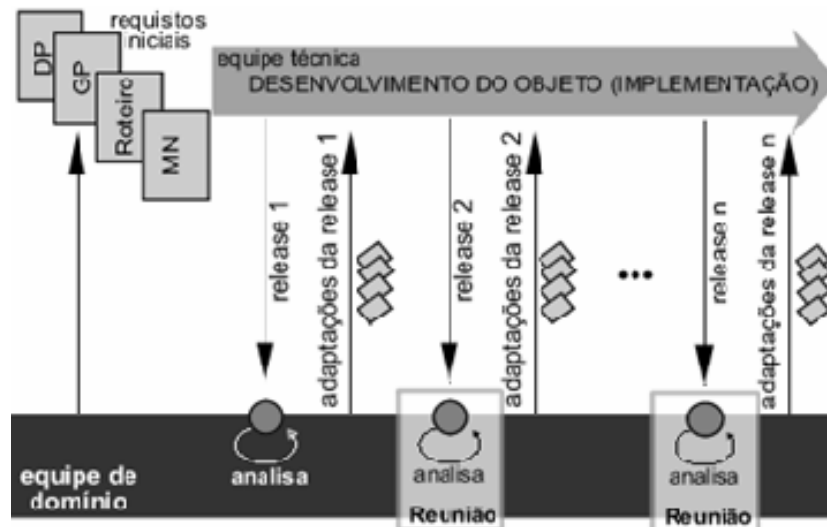
Fonte: (AMARAL et al., 2006)

### 2.1.1.5 Modelo de Processo Colaborativo

O Processo Colaborativo é uma adaptação do processo RIVED, no qual são introduzidas ferramentas baseadas em mapas conceituais, de cenários e navegacionais (CORDEIRO et al., 2007). Neste processo, se faz uso de uma estratégia ágil e colaborativa de desenvolvimento, com a intenção de minimizar os problemas de comunicação e melhorar a compreensão dos requisitos. Desta forma, o trabalho multidisciplinar sendo aplicado em todas as fases com a filosofia de desenvolvimento ágil, reduz o tempo gasto com o desenvolvimento de um OA, pois enquanto um *release* está sendo analisado pela equipe de domínio, outras releases continuam sendo lançadas como melhoramento das anteriores. Na Figura 2.5 é apresentado o modelo colaborativo.

<sup>3</sup><http://www.proativa.virtual.ufc.br/>

Figura 2.5: Modelo de Processo Colaborativo



Fonte: (CORDEIRO et al., 2007)

#### 2.1.1.6 Modelo de Desenvolvimento Baseado na Aprendizagem Significativa

Em Monteiro et al. (2006), é apresentado uma metodologia de desenvolvimento baseada na aprendizagem significativa de Ausubel. O processo de desenvolvimento apresentado na Figura 2.6, foi organizado para permitir o desenvolvimento de OA estruturados em três componentes principais: textos eletrônicos, mapas conceituais e animações interativas. Este por sua vez, procura enfatizar o paralelismo entre as equipes envolvidas, porém ao mesmo tempo, por se tratar de um trabalho bastante multidisciplinar, ressalta a importância da integração dos indivíduos nas tomadas de decisões e verificação de detalhes que possivelmente poderiam ter passado despercebidos durante o desenvolvimento.

Além do modelo de processo em si, também é apresentado em Monteiro et al. (2006) um *framework* de suporte ao desenvolvimento de OA. Este framework foi desenvolvido depois da constatação dos autores de que um determinado número de OA havia uma constante repetição na utilização de alguns componentes, visuais e funcionais.

O framework desenvolvido pelos autores é composto pelos seguintes componentes:

- Loading: Responsável por carregar os elementos gráficos e textuais presentes na animação.

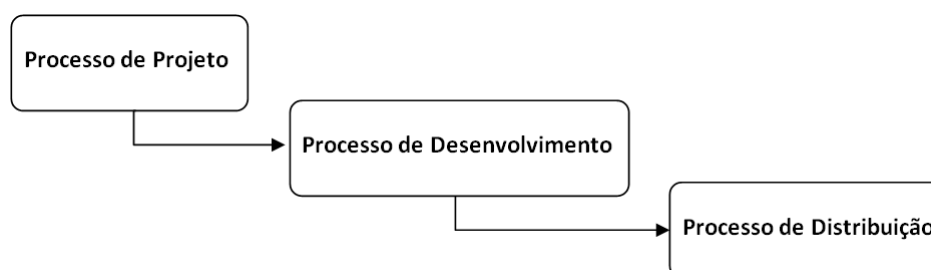
Figura 2.6: Modelo de Desenvolvimento de um OA baseado na Aprendizagem Significativa

1	Definição do tema e contexto do objeto de aprendizagem	
2	Protótipo 1 (fenômeno físico)	Desenvolvimento do Mapa Conceitual
3	Protótipo 2 (contextualização)	Desenvolvimento de textos
4	Integração do Mapa Conceitual	Desenvolvimento de textos
5	Integração dos Textos	Verificação de erros conceituais
6	Verificação de erros e ajustes finais	
<input type="checkbox"/> Equipe de Informática <input type="checkbox"/> Equipe de Física <input checked="" type="checkbox"/> Equipes de Informática e Física		

Fonte: (MONTEIRO et al., 2006)

- Gráficos: Ferramentas didáticas fundamentais para a percepção qualitativa do fenômeno observado.
- Vetores: Elementos fundamentais na descrição das grandezas vetoriais.
- Personagens: Elementos indispensáveis à contextualização do tema abordado.
- Painel de Ajuda: Componente fundamental em qualquer animação de caráter pedagógico, pois fornece instruções de uso e respostas às dúvidas mais frequentes dos usuários.
- Painel de Créditos: Tela onde estão presentes informações detalhadas a respeito de cada colaborador envolvido no desenvolvimento do OA.
- Botões: Componente que inclui os principais elementos gráficos para uma interação agradável e intuitiva com o usuário.
- Mapa Conceitual: Componente que facilita a implementação dos mapas conceituais interativos.

Figura 2.7: Etapas do Processo Sophia



Fonte: Adaptado de Pessoa e Benitti (2008)

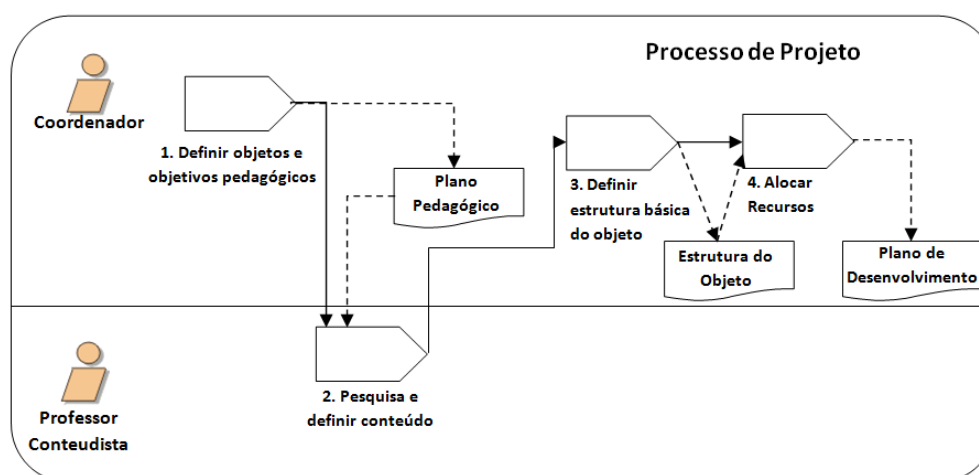
### 2.1.1.7 SOPHIA

Neste processo é destacada a presença de uma fase explícita de distribuição do OA, como mostra a Figura 2.7.

Pessoa e Benitti (2008) descrevem os sub-processos que compõem o Sophia da seguinte forma:

**Projeto:** a meta desta etapa é a definição detalhada da estrutura do OA, envolvendo definição de mídias e conteúdos, bem como o planejamento das atividades necessárias para seu desenvolvimento, alocando recursos e definindo cronograma (detalhando atividades / responsável / prazos). Conforme apresentado na Figura 2.8, os responsáveis pela execução desta etapa são o coordenador e o professor conteudista.

Figura 2.8: Projeto de OA no Processo Sophia



Fonte: Adaptado de Pessoa e Benitti (2008)

Os artefatos produzidos na etapa de projeto devem possuir os seguintes dados fornecendo assim o detalhamento necessário para a equipe de produção do objeto:

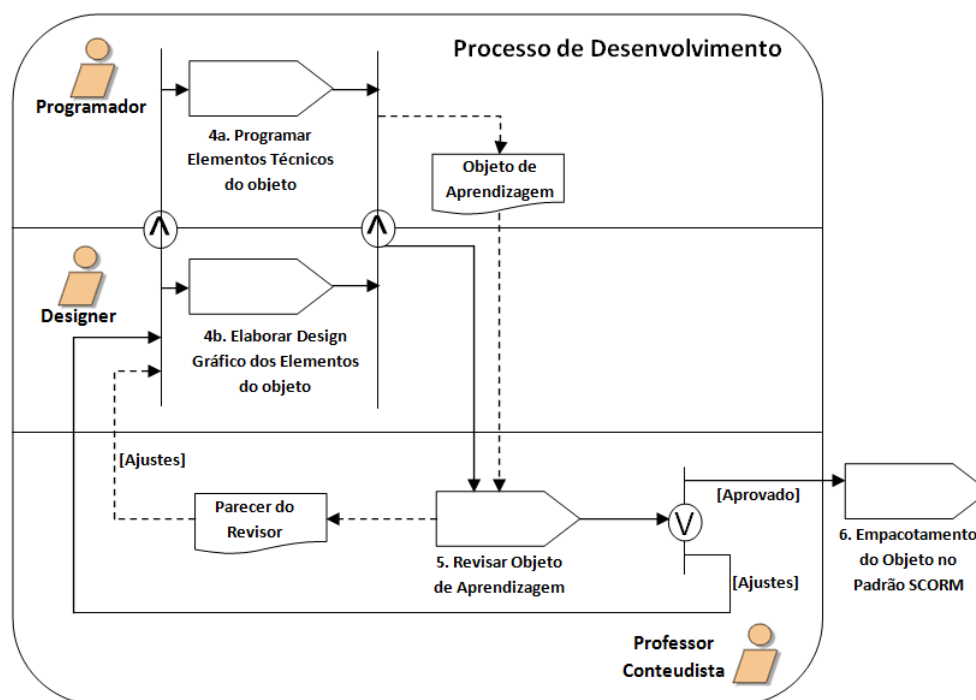
**Plano Pedagógico:** propósitos, objetivos, conceitos presentes, metodologia a ser desenvolvida.

**Estrutura do OA:** tipos de atividades, conceitos presentes, tipos de mídia, visualização das paginas a serem disponibilizadas aos alunos.

**Plano de desenvolvimento:** alocação dos membros da equipe assim como, definição das suas respectivas tarefas e prazos para desenvolvimento.

**Desenvolvimento:** Objetivo desta etapa é produzir o OA, conforme definido na etapa de projeto, deixando-o pronto para disponibilizá-lo aos alunos. Conforme ilustrado na Figura 2.9, esta etapa contempla ao menos uma revisão de todo o OA por parte do professor conteudista. Além disso, pode-se observar que o processo propõe que a produção do OA envolva os papéis de programador e designer, visando imprimir qualidade técnica e de usabilidade ao objeto.

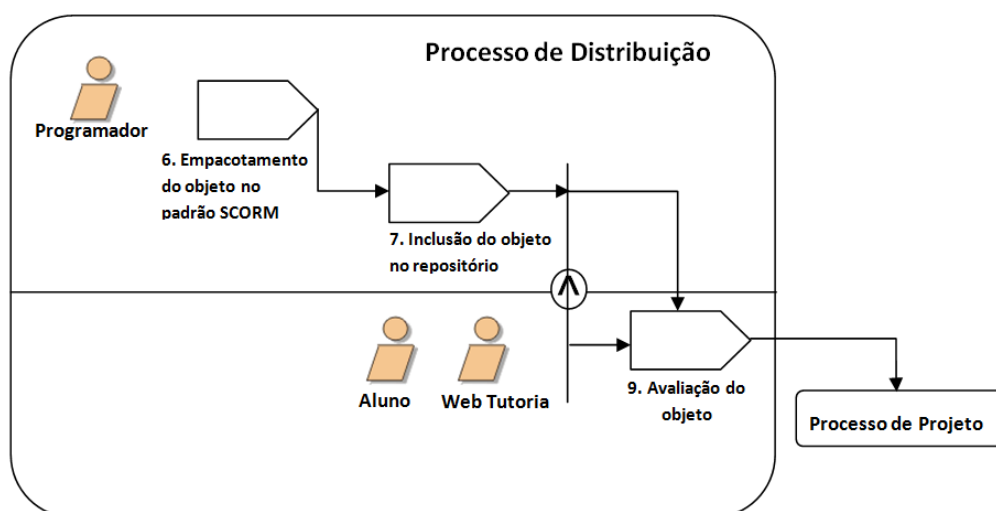
Figura 2.9: Desenvolvimento de OA no Processo Sophia



Fonte: Adaptado de Pessoa e Benitti (2008)

**Distribuição:** Etapa que contempla a disponibilização do OA no repositório observando o padrão SCORM. O processo prevê ainda, conforme ilustrado na Figura 2.10, a avaliação do objeto pela *WebTutoria* e os alunos, visando obter *feedback* para melhoria contínua.

Figura 2.10: Distribuição de OA no Processo Sophia



Fonte: Adaptado de Pessoa e Benitti (2008)

### 2.1.1.8 DART

No âmbito internacional um dos processos de desenvolvimento empregados para o desenvolvimento de OA é o DART. Esse modelo recebe esse nome porque sua origem é decorrente do Projeto *Digital Anthropological Resources for Teaching* (DART) (BOND; INGRAM; RYAN, 2008). Esse modelo é composto por seis estágios, sendo que no primeiro estágio o desenvolvimento é realizado apenas com o professor, de modo a entender as atividades relacionadas ao ensino. No segundo estágio, um protótipo é criado pela equipe técnica, sendo que nos estágios seguintes esse protótipo é avaliado pelos estudantes e posteriormente refinado e disponibilizado para acesso.

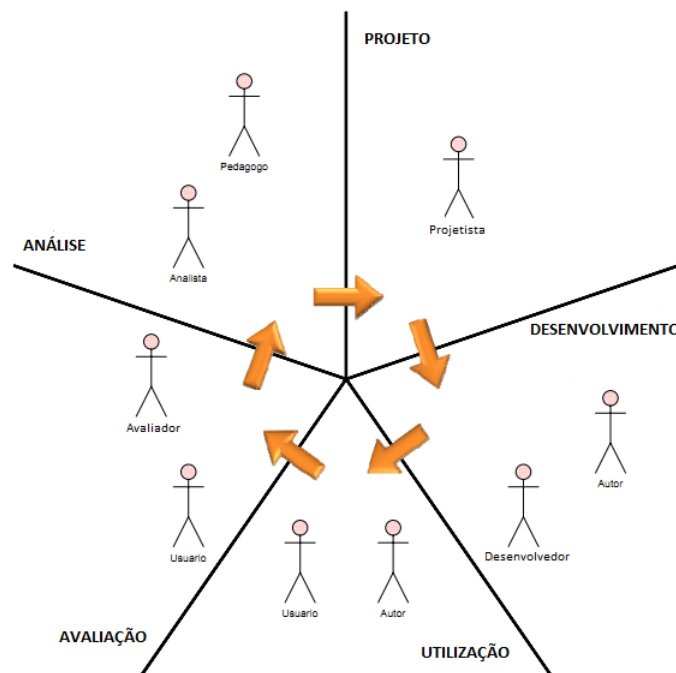
### 2.1.1.9 MIDOA

Ainda no âmbito internacional, é possível citar o Modelo Instrucional para o Desenvolvimento de Objetos de Aprendizagem *Modelo Instruccional para el Desarrollo de Objetos de Aprendizaje* (MIDOA).

Este modelo, segundo Barajas, Muñoz e Álvarez (2007), é estruturado de forma espiral, e cobre cinco fases do desenvolvimento de um OA (Figura 2.11), a saber:

- **Análise:** nesta fase é analisado o projeto instrucional que se quer seguir. Esta atividade é realizada pelo pedagogo e pelo analista cujo objetivo é estabelecer as

Figura 2.11: Modelo de Desenvolvimento MIDOA



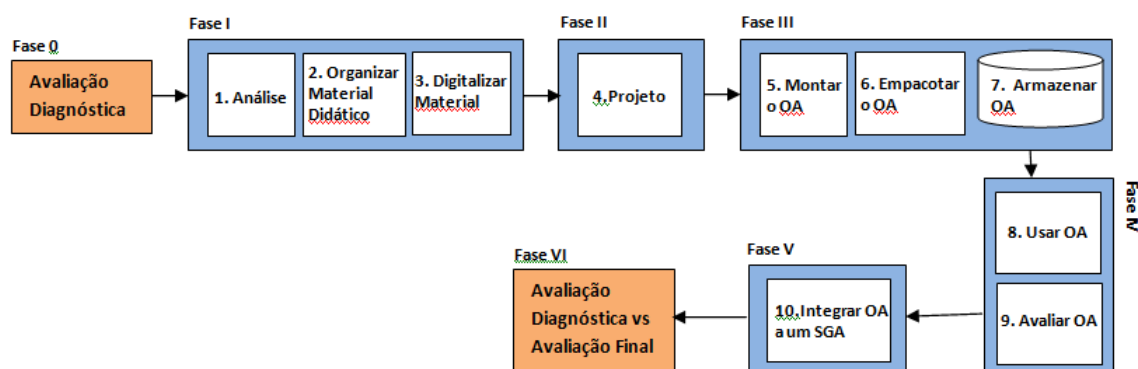
Fonte: Adaptado de Barajas, Muñoz e Álvarez (2007)

atividades de aprendizagem e os mecanismos para se garantir o correto aprendizado.

- **Projeto:** o padrão de desenvolvimento do OA é definido nesta fase. Para tanto, são estabelecidos, de maneira clara, as atividades a serem seguidas, os elementos contextuais e os recursos de informação que serão utilizados para a construção do OA.
- **Desenvolvimento:** os autores devem, nesta fase, contribuir com os recursos de informação em formato digital, a fim de possibilitar a geração do OA. Neste momento, o desenvolvedor irá “colar” os recursos juntos usando ferramentas tecnológicas, como HTML, XML, entre outras. No final desta fase, os OA são instalados em um repositório.
- **Utilização:** nesta fase os OA são utilizados pelos estudantes e professores.
- **Avaliação:** com a utilização dos OA um conjunto de avaliações é gerado, possibilitando a introdução dos produtos.



Figura 2.12: Modelo de Desenvolvimento da UAA



Fonte: Adaptado de GUERRERO, GAONA e RODRÍGUEZ (2007)

### 2.1.1.10 Metodologia de Desenvolvimento da UAA

O processo denominado de Metodologia de Desenvolvimento da UAA foi desenvolvido na *Universidad Autónoma de Aguascalientes* (UAA) e propõe a utilização de uma estratégia ágil (GUERRERO; GAONA; RODRÍGUEZ, 2007). Originalmente, era composto por cinco fases, mas no entanto, outras duas fases foram acrescentadas ao processo, sendo uma no início e a outra no final do mesmo, como mostra a Figura 2.12.

A metodologia de desenvolvimento da UAA é composta pelas seguintes fases (GUERRERO; GAONA; RODRÍGUEZ, 2007):

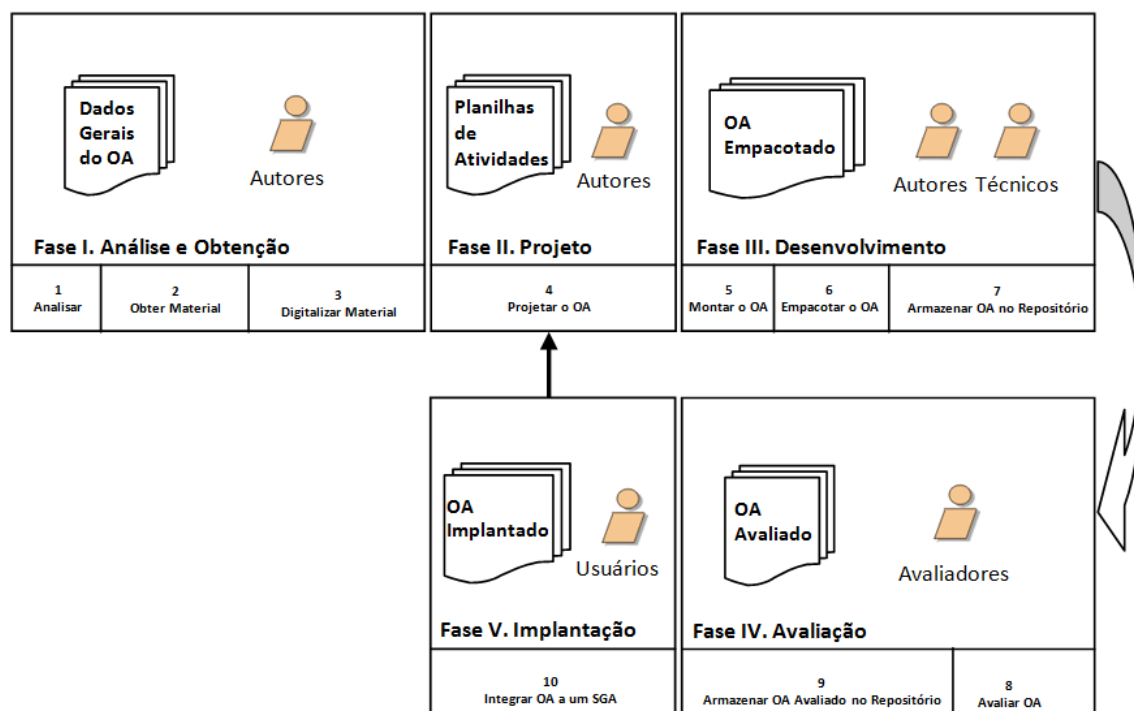
- **Exame Diagnóstico:** antes de se iniciar a construção do OA, esta metodologia indica a determinação do conteúdo a ser desenvolvido através de Redes Semânticas. Em seguida os autores indicam a aplicação de uma avaliação para se verificar o nível de conhecimento da turma sobre o assunto.
- **Análise:** nesta etapa é definida a granularidade do OA. Nela, serão reunidos todos os recursos que irão ser utilizados no OA.
- **Projeto:** aqui será realizado o projeto do OA. Esta etapa deverá gerar como resultado uma versão inicial da aparência do OA.
- **Desenvolvimento:** se faz uso de uma ferramenta *web* para empacotamento, montagem e integração da OA em um repositório.

- **Avaliação:** Nesta etapa, o OA será avaliado tanto técnica quanto pedagogicamente.
- **Implementação:** Aqui, a OA é aplicado no contexto educacional.
- **Avaliação Comparativa:** Nesta fase, se aplica uma avaliação após a utilização do OA e os resultados alcançados são comparados com os gerados na avaliação diagnóstica, para estabelecer um indicador, que servirá como um parâmetro para realizar uma análise da função pedagógica do recurso.

### 2.1.1.11 OADDEI

Outro processo de desenvolvimento de OA é o descrito em Arteaga et al. (2006). O processo proposto neste trabalho, denominado AODDEI, tem como base o modelo de desenvolvimento instrucional ADDIE (SMITH; RAGAN, 1999) e é composta por cinco fases, como apresentado na Figura 2.13.

Figura 2.13: Metodologia AODDEI



Fonte: Adaptado de Arteaga et al. (2006)

Cada fase, por sua vez, é formada por um conjunto de passos, que descrevem atividades dentro de cada fase. De forma resumida, a metodologia apresenta a seguinte

estrutura:

- Fase I - Análise e Obtenção
  - Passo 1 - Análise: nesta atividade são descritas as informações relativas ao contexto do OA, assim, são descritas várias informações relacionados aos objetos, tais como: nome do OA, Descrição do OA, nível de escolaridade, objetivos de aprendizagem, entre outras.
  - Passo 2 - Obtenção de Material: os materiais didáticos relacionados ao OA são obtidos nesta atividade.
  - Passo 3 - Digitalizar o Material: este passo é executado somente nos materiais didáticos que não estiverem em formato digital.
- Fase II - Projeto
  - Passo 4 - Estruturação do OA: o projeto do OA nesta metodologia está ligado à definição das relações entre os objetivos pedagógicos do OA, os seus recursos e as atividades a serem empregadas para que os recursos sejam eficientes no alcance dos objetivos
- Fase III - Desenvolvimento
  - Passo 5 - Montagem: utilizando a estrutura de projeto da fase anterior e os recursos a serem utilizados, é construído um arquivo HTML com a estrutura organizada do OA.
  - Passo 6 - Empacotamento: aqui são criados os metadados do OA no formato SCORM.
  - Passo 7 - Instalação do OA em um repositório temporário: o OA é instalado em um repositório temporário para fins de avaliação.
- Fase IV - Avaliação
  - Passo 8 - Avaliação do OA: O OA é avaliado por um conjunto de *experts*, considerando aspectos como projeto, conteúdo e metadados.

- Passo 9 - Instalação do OA: O OA é instalado em um repositório para ser utilizado.
- Fase V - Implantação
  - Passo 10 - Integração do OA a um Ambiente Virtual de Aprendizagem (AVA): O OA é integrado a um AVA, como o MOODLE, com a finalidade de dar suporte às atividades à distância.

#### 2.1.1.12 LOCoMo

Um outro trabalho que segue uma estrutura complexa de gerenciamento de projetos para a construção de OA é o apresentado em Balda (2008). Neste trabalho o autor apresenta a metodologia *Learning Objects Construction Methodology* (LOCoMo). Essa metodologia é baseada no RUP, e é composta das seguintes fases:

- **Análise:** nesta fase se estabelece a metáfora a ser utilizada no OA e as características detalhadas do objeto. Nesta são definidos claramente os requisitos tanto em nível conceitual como funcional. Esta fase tem como objetivos a definição de casos de uso, a estimativa de custo e tempo de desenvolvimento, a criação do plano de projeto, entre outros.
- **Projeto Conceitual:** uma vez estabelecido os objetivos do OA se procede nesta fase a descrição de como estes objetivos serão atingidos. Durante esta fase são desenvolvidos modelos de dados e da aplicação, que geram artefatos tais como a arquitetura do software e o modelo de casos de uso.
- **Construção:** nesta fase o OA é desenvolvido. Esta fase é composta por diversas subfases, a saber:
  - **Desenvolvimento de recursos:** são criados os recursos a serem usados no OA, tais como botões, arquivos, imagens, entre outros.
  - \* **Adequação aos padrões:** todos os recursos desenvolvidos são associados de forma a gerar a aplicação seguindo os objetivos do OA. Nesta fase também são gerados os metadados associados ao OA.

- \* Avaliação Pedagógica: os objetivos pedagógicos do OA são avaliados nesta fase.

De modo geral, todos os processos apresentados descrevem etapas de desenvolvimento bastante semelhantes. Particularmente, são descritas etapas relacionadas ao projeto pedagógico, seguidas de etapas de projeto gráfico e da implementação e instalação dos OA em repositórios.

Além disso, esses modelos também apresentam estruturas bastante semelhantes no que se refere à organização das equipes de desenvolvimento. Inicialmente a multidisciplinaridade é levantada em todos os trabalhos apresentados. Também em algumas metodologias (BARAJAS; MUÑOZ; ÁLVAREZ, 2007), (PESSOA; BENITTI, 2008), (ARTURO et al., 2009), (ARTEAGA et al., 2006) e (BALDA; LÓPEZ, 2008) são descritos papéis e atividades para cada um dos diferentes profissionais que participam do desenvolvimento de um OA.

Com base no quantitativo de processos de desenvolvimento descritos nesta subseção, será apresentada a seguir, uma análise comparativa desses processos com vista, dentre outras coisas, na metodologia e padrão de desenvolvimento adotados.

### 2.1.2 Análise Comparativa dos Processos de Desenvolvimento

Para dar início à análise comparativa dos processos empregados na produção de objetos de aprendizagem, é apresentado no Quadro 2 um paralelo entre as principais características comuns a esses processos.

No Quadro 2.1, são utilizadas as siglas NE, DP e GP, que significam, respectivamente, não especificado, *design* pedagógico e guia do professor. O NE é utilizado sempre que não tiver sido explicitado na literatura a existência do critério analisado. Já o DP (*Design* Pedagógico) e o GP (Guia do Professor) são documentos produzidos durante a fase de concepção e projeto, respectivamente, e servem para mediar a comunicação entre os membros da equipe, norteando-os na produção do OA .

Os processos descritos nesta tese, em sua maioria, fazem uso de alguma metodologia de desenvolvimento (cascata, incremental ou espiral) como forma de melhorar a

qualidade da documentação gerada e otimizar o tempo de desenvolvimento do recurso. No entanto, ainda é possível observar que não há relatos na literatura, de utilização de documentos formais por parte da equipe de desenvolvimento. Essa inexistência, acaba por interferir em outro aspecto importante a ser levantado, dentro da esfera de processos de desenvolvimento, que é o fator comunicação entre os membros da equipe multidisciplinar.

Com relação aos processos apresentados, é comum observar a utilização de documentos, utilizados pela equipe de desenvolvimento, escritos em linguagens naturais como por exemplo a utilização do DP, GP e *Storyboard* (roteiro). Como as equipes são, geralmente, compostas por pedagogos, especialistas de área (*experts*), designers e programadores, a documentação gerada em linguagem natural compromete a qualidade e o tempo de desenvolvimento do recurso, uma vez que esse tipo de documentação (escrito em linguagem natural) é passível de dupla interpretação. Muito embora esse tipo de documentação não contemple uma boa comunicação entre os membros de uma equipe multidisciplinar, nem tão pouco favoreça a otimização do tempo de desenvolvimento, ainda não se chegou a um consenso sobre o tipo de documentação a ser utilizado de modo que a mesma pudesse ser compreendida pelos diferentes profissionais envolvidos no desenvolvimento do recurso.

É possível observar também que alguns processos são derivados de outros processos. Essas derivações são decorrentes de ajustes realizados no processo de origem, que visam adaptá-lo ao tipo de recurso a ser desenvolvido como por exemplo os processos Proativa e Colaborativo que foram criados a partir de ajustes realizados no processo Rived.

Vale ainda ressaltar, que o processo ADDIE serviu de base para a criação de uma grande parte dos processos de desenvolvimento atualmente existentes. Isto se deve ao fato deste processo ter como objetivo a produção de material instrucional e para isto era necessário planejar, projetar e avaliar os ganhos que o recurso, uma vez produzido, poderia proporcionar.

Outro ponto a ser levado em consideração e que vem se tornando prática comum, é o fato dos recursos, após desenvolvidos, serem armazenados em repositórios para serem utilizados ou reutilizados. No entanto, é possível observar que o tipo de reuso

Quadro 2.1: Análise Comparativa entre os Processos de Desenvolvimento de RED

Processo	Metodologia	Documentação	Adaptatividade	Validação
ADDIE	NE	NE	NÃO	Professores e Alunos
Labvirt	NE	<i>Storyboard</i>	NÃO	Alunos
Rived	NE	DP, <i>Storyboard</i> e GP	NÃO	NE
Proativa	NE	DP, <i>Storyboard</i> e GP. Paralelismo entre as equipes	NÃO	Professores e Alunos
Colaborativo	Ágil e Colaborativa	DP, <i>Storyboard</i> e GP. Produz <i>releases</i> . Preocupação com a comunicação entre os membros da equipe	NÃO	NE
Baseado na Aprendizagem Significativa	NE	NE. Enfatiza paralelismo entre as equipes. Criação de um <i>framework</i>	NÃO	Professores e Alunos
Sophia	NE	NE. Enfatiza paralelismo entre as equipes	NÃO	Professores
DART	NE	NE. Não contempla paralelismo entre as equipes	NÃO	Alunos
MIDOA	NE	NE	NÃO	Professores e Alunos
UAA	NE	NE	NÃO	NE
OADDEI	NE	NE	NÃO	<i>Experts</i> nas áreas nas quais os recursos foram desenvolvidos
LOCoMo	NE	NE	NÃO	Atores não especificados

Fonte: Própria Autora

permitido é limitado, haja visto que nenhum dos processos apresentados, nesta tese, contempla a característica da adaptatividade, conforme apresentado no Quadro 2.1.

Devido ao tipo de material produzido, a ausência da adaptatividade acaba por limitar a autonomia docente, no que concerne as modificações na interface do recurso realizadas pelos próprios professores.

Dessa forma, será apresentado na seção 2.2 a relevância em poder adaptar um recurso, bem como as técnicas mais utilizadas para adaptá-los.

## 2.2 Adaptação de Recursos Educacionais Digitais

O crescente uso da internet juntamente com o processo de globalização favoreceu a disseminação do uso dos recursos educacionais digitais e sua utilização por um número maior de usuários. Recursos esses, que independente da localidade em que foram desenvolvidos, podem ser utilizados ou aplicados em diversos cenários, bastando apenas que os mesmos passem por um processo de adaptação para poder, caso necessário, adequar-se às novas necessidades relacionadas ao perfil, à cultura ou ao próprio nível de conhecimento do aprendiz.

O termo adaptação de software surgiu com o propósito de fornecer técnicas para fazer arranjos em peças de software já desenvolvidos, a fim de reutilizá-los em novos sistemas (CANAL et al., 2006). Sendo assim, a adaptação pode ser definida como qualquer alteração na estrutura ou funcionamento de um organismo, passível de adequá-lo melhor ao seu ambiente (DICTIONARY, 1989).

A adaptação pode ser realizada tanto dinâmica quanto estaticamente (ROCHA et al., 2007). A primeira delas ocorre no nível de sistema, nesse caso recebe o nome de personalização. Já a segunda ocorre no nível de interface, recebendo o nome de customização (FREUND, 2008).

A personalização não deve ser confundida com a customização, pois enquanto a customização refere-se à mudança, montagem ou modificação de um produto ou serviço de acordo com as necessidades dos clientes, a personalização envolve intensa comunicação



e interação entre as duas partes, ou seja, cliente e fornecedor (FREUND, 2008).

A compreensão da autora sobre customização e personalização foi facilitada devido ao trabalho desenvolvido junto ao Grupo de Redes de Computadores Engenharia de Software e Sistemas (GREat) da UFC, que tem experiência na adaptação (personalização) de software para dispositivos móveis (celulares).

Em meio à discussão sobre as possíveis formas de adaptação de um RED, será descrito na seção 2.2.1 técnicas de adaptação adotadas para a adequação do recurso ao público.

### 2.2.1 Técnicas de Adaptação

Com base na quantidade de RED desenvolvidos e da importância em reutilizá-los, é apresentado nesta seção alguns dos resultados encontrados na literatura que descrevem sobre a personalização e customização de recursos educacionais digitais como técnicas empregadas na adaptação de materiais educacionais.

No contexto da personalização, as técnicas de internacionalização e localização primam por aproximar o maior número de pessoas (PRUDÊNCIO; VALOIS; LUCCA, 2005) e estão relacionadas à adaptação de software.

Segundo PRUDÊNCIO, VALOIS e LUCCA (2005), a internacionalização (também conhecida como I18N) é uma etapa do processo de desenvolvimento do software que deve fazê-lo flexível e neutro em termos de relações culturais, financeiras e legais de um país. Um software internacional deve admitir, por exemplo, distintas formatações de números e algoritmos de ordenação que sigam as diferentes regras dos diversos idiomas. Para implementar estas características é necessário utilizar procedimentos adequados, para que seja possível localizar o software apropriadamente. Já a localização (também conhecida como L10N) é o segundo passo na preparação de um software para o mercado internacional, é neste momento em que se trata da adaptação do mesmo para um local específico em que ele será utilizado. Envolve tradução, adaptação cultural, adaptação a normas e características do mercado-alvo.

Na Figura 2.14 é possível observar a relação de dependência existente entre essas

duas técnicas. Em outras palavras, só é possível realizar a localização de um software se ele for internacionalizável.

Figura 2.14: Relação de Dependência entre Internacionalização e Localização de Software.



Fonte: Própria Autora

Segundo a Associação de Padronização do Setor da Localização (LISA), tornar um software localizável envolve pegar um produto e torná-lo lingüística e culturalmente adequado à localidade de destino (país/região e idioma), onde ele será utilizado (PRUDÊNCIO; VALOIS; LUCCA, 2005).

Apesar da localização ser uma alternativa de aproximar o software do usuário, para Amiel, Squires e Orey (2009) e Dagiene e Zilinskiene (2009) essa estratégia não é adequada quando seu foco é somente na tradução do idioma. Isso porque se pensarmos em OA, o contexto do design é muito importante e, portanto, deve ser levado em consideração.

Segundo Amiel, Orey e West (2010), essa importância ocorre porque é possível se observar a existência de uma estreita relação, entre os elementos culturais e o design, de um recurso educacional. Ainda conforme Amiel, existem três variantes dos recursos educacionais que têm como objetivo salientar os elementos culturais, conhecidas como Recursos Educacionais com Saliências Culturais (RESC), n-Cultura e o Recursos Educacionais com Adaptações Culturais (REAC). A primeira alternativa busca expandir o conceito de RE ao explicitar os elementos culturais além de oferecer suporte e explicações (por exemplo: guia para o professor ou aprendiz), já as duas últimas alternativas contemplam a adaptação.

Para Dagiene e Zilinskiene (2009), a localização quando realizada em recursos de matemática, deve focar sua atenção, principalmente, na representação dos caracteres matemáticos, pois estes diferem bastante. Exemplo disso, são as operações aritméticas de multiplicação e divisão que têm sua notação diferenciada de país para país, conforme apresentado na Quadro 2.2.

Quadro 2.2: Notação Matemática em Diferentes Países

		NOTAÇÃO MATEMÁTICA					
OPERAÇÃO MATEMÁTICA	MULTIPLICAÇÃO	<b>ÁUSTRIA</b> $\begin{array}{r} 1014 \cdot 163 \\ 6084 \\ \underline{3042} \\ 165282 \end{array}$					
	DIVISÃO	<b>ÁUSTRIA</b> $9163:14=654$ 076 063 7R	<b>DINAMARCA</b> $5/375 \setminus 75$ $\frac{35}{25}$	<b>ISRAEL</b> $\frac{75}{375 \mid 5}$ $\frac{35}{25}$ $\frac{25}{00}$	<b>JAPÃO</b> $\frac{75}{5) 376}$ $\frac{35}{26}$ $\frac{25}{1}$ RESPOSTA : 75...1	<b>CROÁCIA</b> $376:5=75,2$ $\frac{35}{26}$ $\frac{25}{10}$	<b>AUSTRÁLIA</b> $\frac{174 \text{ r}1}{5) 8^3 7^2 1}$

Fonte: Adaptado de Dagiene e Zilinskiene (2009)

Já na perspectiva de customização, surge a possibilidade do próprio usuário modificar um determinado cenário no intuito de adequá-lo ao seu novo contexto. Desse modo, algumas iniciativas na customização de OA serão apresentadas.

Para NIELSEN (2009), a customização pode ocorrer tanto em nível de interface, quanto em nível de produto. Nessa primeira é permitido aos utilizadores personalizar a sua experiência *on-line* através da adaptação da interface do usuário de acordo com suas preferências. Enquanto a segunda possibilita a configuração da funcionalidade do recurso.

Em Coner (2003), a customização ocorre a partir da implementação de metadados que ajudam na identificação dos estilos de aprendizagem presentes em um recurso educacional. Essa medida auxilia na recuperação de objetos de aprendizagem em repositórios levando em consideração o perfil e a experiência do aprendiz.

Já em Gkatzidou e Pearson (2009), a customização de *web* sites educacionais é

tratada de modo que o conteúdo seja categorizado para atender ao perfil do usuário.

Segundo Kahle (2008) é importante projetar tecnologias adaptativas e flexíveis que minimizem a sobrecarga organizacional e técnica atribuída a estudantes e educadores, possibilitando assim, que eles assumam o centro do controle e aproveitem de fato, o potencial transformador da tecnologia para a educação.

De fato, a customização proporciona uma intervenção direta do usuário (professor) no sistema e isto acaba incidindo sobre o nível de autonomia do mesmo.

Nesse contexto de se customizar recursos educacionais digitais e levando em consideração as diferentes visões e importância da customização, é conduzida na seção 2.2.2 uma revisão da literatura sobre as perspectivas de construção e adaptação desses materiais com o auxílio de ferramentas de autoria.

## **2.2.2 Ferramentas de Autoria**

De acordo com NIELSEN (2009), os professores necessitam de ferramentas para desenvolver sua independência na criação de ambientes que fascinem e motivem os estudantes. Essa independência ou autonomia para desenvolver tais ambientes, com o tempo, irá ajudar professores a inovar na criação de diferentes oportunidades de aprendizagem.

Diante da necessidade de ampliação da autonomia docente no que concerne a criação de RED, são apresentados nesta seção exemplos de ferramentas de autoria que buscam viabilizar a criação e produção de objetos de aprendizagem, sem a necessidade de suporte técnico.

Na literatura é possível encontrar descrições de experiências no desenvolvimento de ferramentas de autoria para ajudar na construção de RED do tipo OA. O resultado de algumas dessas experiências podem ser visualizados com a descrição, nas próximas seções, das ferramentas MLOAT, GLO Maker, LOC e RELOAD.

Figura 2.15: Tela Principal da Ferramenta de Autoria MLOAT

The screenshot shows the 'Admin Page' of the MLOAT tool. At the top, there is a header with the text '3 Way Tool the FLV version Authoring Interface'. Below the header, the page is titled 'Admin Page' and contains the following instructions:

This configuration console provide designers several tools in configure and control the multimedia learning object which was just created.

Please bookmark this page for future references.

1. Write down your multimedia learning object key number

597570521

2. Upload movie and image files.

The 'Upload Flash Video' section includes a text box with the instruction: 'Upload your Flash Video(\*.flv) file here. A Flash Video(\*.flv) file is required to generate the timeline before the next step: synchronization. (Only one \*.flv file is saved on the server, every new video upload will erase the old video file. )'. Below the text box are two buttons: 'Seleccionar archivo...' and 'Upload File'. To the right of the text box, it says: 'Your video directory: Directory is empty. Please upload image files.'

The 'Upload Images' section includes a text box with the instruction: 'Upload your image files here. Valid image formats are \*.jpg, \*.gif and \*.swf.'. Below the text box are two buttons: 'Seleccionar archivo...' and 'Upload File'. To the right of the text box, it says: 'Your image directory: Directory is empty. Please upload movie and image files.'

3. Click [here](#) to configure the synchronization

Fonte: Própria Autora

### 2.2.2.1 *Multimedia Learning Object* (MLOAT)

O *Multimedia Learning Object Authoring Tools* (MLOAT)<sup>4</sup> é uma ferramenta de autoria *open source* desenvolvida pela Universidade da Columbia Britânica e tem por objetivo fazer a composição de diversos recursos como por exemplo, vídeo, áudio, imagens e textos, transformando-os em um novo recurso. Na Figura 2.15, pode-se observar a tela principal do MLOAT. Nela o usuário tem três passos a seguir, sendo o primeiro deles o fornecimento do número chave que o usuário da ferramenta deve anotar como forma de localizar mais facilmente o novo recurso a ser criado. O segundo passo consiste em inserir um vídeo, seguido pela inserção de uma imagem, ambos no formato especificado pela ferramenta. Já o terceiro e último passo, consiste na sincronização dos recursos anexados.

Ao sincronizar os recursos, uma tela contendo vídeo e imagem é fornecida, conforme Figura 2.16. Nela, existe ainda um campo para, caso necessário, realizar a inserção de texto. Esse texto pode ser inserido em todo o vídeo ou em um trecho específico. Feito

<sup>4</sup><http://www.learningtools.arts.ubc.ca/mloat.htm>

isso, o usuário tem a possibilidade de salvar (no servidor) seu recurso para ser utilizado em sala de aula ou em outro momento.

Figura 2.16: RED criado com a Ferramenta de Autoria MLOAT



Fonte: Própria Autora

Todos os recursos são configurados para serem reproduzidos em uma ordem pré-configurada. Os usuários não precisam executar qualquer tarefa de programação para gerar um objeto de aprendizagem.

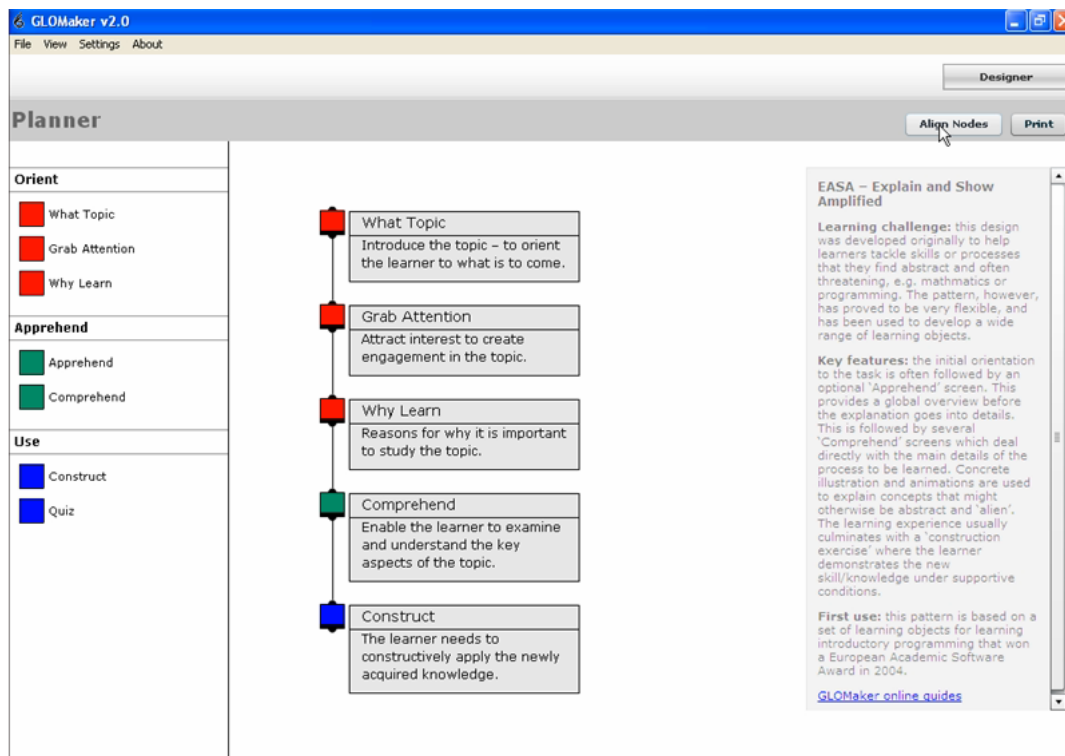
### 2.2.2.2 *Generative Learning Object Maker (GLO Maker)*

Outro exemplo de ferramenta de autoria foi desenvolvido pela Universidade Metropolitana de Londres e denominado de *Generative Learning Object Maker authoring tool* (GLO Maker)<sup>5</sup>. O GLO Maker (Figura 2.17) é uma ferramenta de autoria livre para a criação de objetos de aprendizagem desenvolvida com o intuito de capacitar professores e outros usuários, no desenvolvimento de objetos de aprendizagem multimídia altamente adaptável. A ferramenta tem seu processo de autoria orientado a projeto

<sup>5</sup><http://www.glomaker.org/>

com duas partes principais: o **Planner** onde é construído o enredo básico do projeto de aprendizagem e o **Designer** onde as telas são criadas com base em modelos flexíveis (KOOHANG; FLOYD; STEWART, 2011).

Figura 2.17: Tela Principal da Ferramenta de Autoria GLO Maker

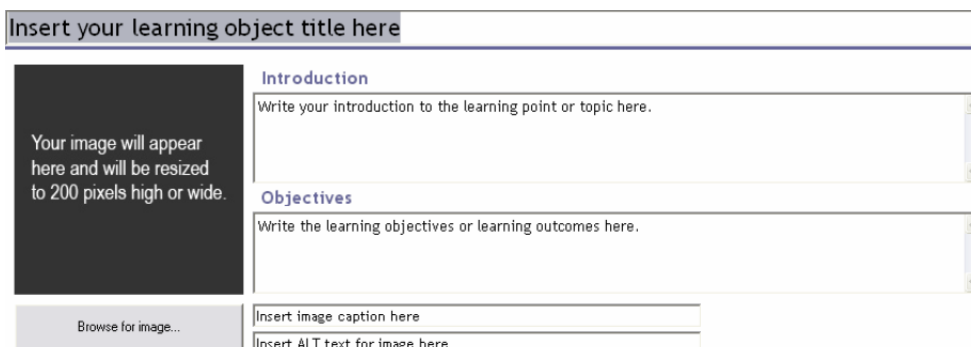


Fonte: Própria Autora

No GLO Maker é possível usar modelos de projeto para estruturar o seu objeto de aprendizagem, ou se o usuário preferir pode usar o modo *Freestyle* para criar seu próprio projeto. Todos os projetos são executáveis - eles permitem que objetos de aprendizagem multimídia sejam criados diretamente e executados na *web* ou em um ambiente virtual de aprendizagem como o MOODLE.

O objeto de aprendizagem criado com o GLO Maker pode ser compartilhado, trocado e adaptado, por qualquer usuário, para atender às necessidades e preferências locais.

Figura 2.18: Tela com visão parcial do LOC mostrando orientação pedagógica para o desenvolvedor



The screenshot displays the Learning Object Creator (LOC) interface. At the top, there is a text input field labeled "Insert your learning object title here". Below this, on the left, is a dark grey box with the text "Your image will appear here and will be resized to 200 pixels high or wide." and a "Browse for image..." button. To the right of the image box are two text input fields: "Introduction" with the instruction "Write your introduction to the learning point or topic here." and "Objectives" with the instruction "Write the learning objectives or learning outcomes here.". Below these are two more text input fields: "Insert image caption here" and "Insert ALT text for image here".

Fonte: Própria Autora

### 2.2.2.3 *Learning Object Creator* (LOC)

Outro exemplo de ferramenta de autoria é o *Learning Object Creator* (LOC)<sup>6</sup>. Esta ferramenta *open source* é projetada para ajudar os professores no processo de criação de seus próprios materiais educativos sem a necessidade de suporte técnico (WATSON; DICKENS; GILCHRIST, 2008).

Desenvolvido pelo Centro de assuntos para as Línguas, Linguística e Estudos Regionais (*LLAS Subject Centre*) da universidade de Southampton. Ela está disponível gratuitamente aos professores de Instituições de Ensino Superior do Reino Unido para fins não comerciais, educacionais, e pode ser usado para uma variedade de áreas.

A LOC (Figura 2.18) é uma ferramenta de código aberto, desde que ela seja usada para fins educacionais (uso comercial exigirá uma taxa de licença a ser paga), mas uma taxa será exigida para a oficina de formação inicial que é obrigatória para ter acesso a ferramenta.

LOC é baseado em atividade e inclui seis diferentes tipos de tarefas interativas, provisão para *feedback* formativo, fazendo referência bibliográfica e a acomodação de uma variedade de formatos de mídia (texto, áudio, vídeo, imagens). É fácil de usar e vem com estilos diferentes, que dá aos professores a oportunidade de criar materiais de aprendizagem (KOOHANG; FLOYD; STEWART, 2011).

Os materiais gerados pelo ferramenta LOC visa integrar métodos pedagógico de

---

<sup>6</sup><http://loc.llas.ac.uk/>



ensino a distância para a fácil criação de materiais online. Os métodos são distribuídos em planejar, editar, revisar e publicar.

O planejar no LOC consiste em apoiar a criação de uma atividade de aprendizagem bem planejada e eficaz. Permite a realização de um planejamento *off-line* que o usuário se concentre em pedagogia, em vez de tecnologia além de possibilitar salvar seu planejamento ao lado do material online.

Já o editar permite que LOC adapte o objeto de aprendizagem, armazene-o online, para que fique acessível de qualquer lugar do mundo além de possibilitar a inclusão de outros arquivos e vídeos.

Com o LOC é possível também publicar o objeto de aprendizagem produzido para que os outros na comunidade possam revisá-los e com isso obter *feedback* para melhoria do recurso.

Após a conclusão do OA, o mesmo pode ser publicado gratuitamente, permitindo que qualquer pessoa, em qualquer lugar no mundo, possa acessá-lo. Os OA são salvos em formato ZIP e pode ser inserido em qualquer site ou ambiente virtual de aprendizagem.

#### 2.2.2.4 *Reusable eLearning Object Authoring & Delivery (RELOAD)*

O *Reusable eLearning Object Authoring and Delivery (RELOAD)*<sup>7</sup> é uma ferramenta de autoria desenvolvida em parceria pelas universidades de Strathclyde e Bolton, ambas situadas no reino unido, sendo atribuída a esta última universidade a responsabilidade pelo gerenciamento da ferramenta.

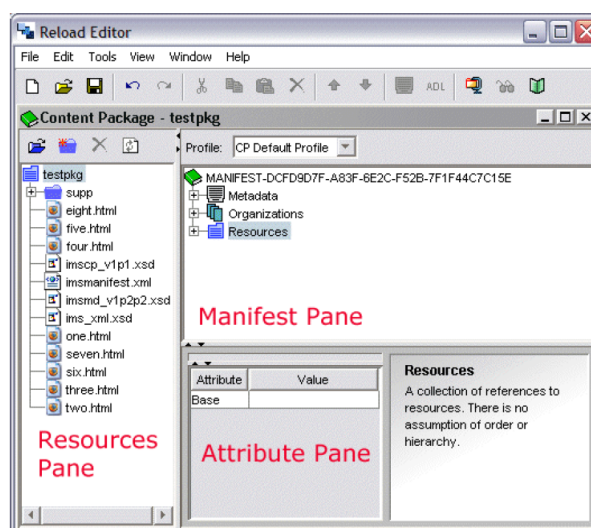
O RELOAD tem dois objetivos principais. O primeiro, consiste em possibilitar a criação, compartilhamento e reutilização de objetos de aprendizagem e serviços, já o segundo, visa aumentar o leque de abordagens pedagógicas realizáveis através da utilização de planos de aula (GRIFFITHS et al., 2007).

O material criado com o suporte desta ferramenta tem o formato de páginas *web* com sequência pré-determinada. Após sua criação, os mesmos podem ser empacota-

---

<sup>7</sup><http://www.reload.ac.uk/>

Figura 2.19: Interface da Ferramenta RELOAD



Fonte: (GRIFFITHS et al., 2007)

dos, compactados e compartilhados com outras pessoas utilizando para isso Ambientes Virtuais de Aprendizagem. O Reload, faz uso do padrão IMS (*System Management Instructional*) que por sua vez tem a preocupação em analisar as diversas abordagens pedagógicas e suas atividades de aprendizagem, buscando alcançar um meio termo entre aplicação pedagógica e um bom nível generalização (DUTRA; TAROUÇO, 2011). O RELOAD a partir da versão 1.3 passa a oferecer também suporte ao padrão SCORM.

Na Figura 2.19 é possível visualizar a interface desta ferramenta e observar que a mesma está organizada em três espaços diferenciados no qual o primeiro é a área chave da representação da estrutura do pacote do conteúdo (*Resources Pane*). A segunda é a área para inserção dos metadados, organização (sequência) e recursos a serem utilizados (*Manifest Pane*). Por fim, a terceira área faz referência a informação sensível ao contexto sobre os elementos selecionados como uma tabela de valores e caixas para edição de atributos (*Attribute Pane*).

### 2.2.2.5 Outras Ferramentas

Xical - Xical<sup>8</sup> é uma ferramenta *Open Source*, desenvolvida em *flash*, que possibilita a construção de apresentações, palestras a distância, tutoriais ou mesmo seminários via *web*. O material é gerado no formato *media player*. É designada e programada em

<sup>8</sup><http://xical.org>

*flash* e *actionScript* e pode ser executada em qualquer sistema operacional que forneça suporte ao *flash* (KOOHANG; FLOYD; STEWART, 2011).

eXe Learning<sup>9</sup> é uma ferramenta de autoria *Open Source* destinada a ajudar professores e acadêmicos na criação de conteúdos educacionais *web*, sem a necessidade, que os mesmos, tenham conhecimento em XML ou marcação XML. Os recursos criados podem ser exportados para o formato IMS *Content Package*, SCORM 1.2, IMS *Common Cartridge* ou ainda como simples conteúdo de página *web*.

Pachyderm<sup>10</sup> é uma ferramenta de autoria de fácil utilização, desenvolvida para criação de recursos multimedia. Desenvolvida especialmente para pessoas que tem pouca ou nenhuma experiência na criação multimídia, possibilitando que os usuários façam o *upload* de suas próprias media (imagem, áudio e pequenos segmentos de vídeo) e os coloque em modelos pré-estabelecidos, nos quais inclui a função de construir *player* de vídeo e áudio, podendo também associá-los a outros *templates* ou a outros recursos.

Nesta ferramenta, o texto pode ser copiado e colado, ou inserido diretamente na Pachyderm. Finalizadas as telas e suas interligações, a apresentação é publicada e pode ser baixada e colocada no site do autor, em um CD, ou em outro lugar. Autores podem também, deixar as suas apresentações no servidor Pachyderm com *link* de acesso. O resultado é uma atrativa apresentação em *Flash* baseado em multimídia interativa.

### 2.2.3 Análise Comparativa das Ferramentas de Autoria

Diante do exposto sobre a dificuldade enfrentada por professores no desenvolvimento de seus próprios recursos, será apresentada no Quadro 2.3 uma análise comparativa das diversas ferramentas de autoria existentes na literatura e citadas neste trabalho. Essas ferramentas visam ampliar a possibilidade do professor atuar não somente em parte do processo de desenvolvimento, mas nele como um todo.

Conforme resumido no Quadro 2.3, o recurso produzido com o auxílio das ferramentas apresentadas viabilizam a criação de recursos no formato de slides quer seja através da composição de áudio, vídeos, imagens e textos ou através da criação de

---

<sup>9</sup><http://exelearning.org/wiki>

<sup>10</sup><http://pachyderm.nmc.org/>

Quadro 2.3: Análise Comparativa das Ferramentas de Autoria de RED

Ferramenta de Autoria	Características
MLOAT	Livre para <i>download</i> . Possibilita a criação de recursos a partir da composição de outros recursos, isentando o usuário da tarefa de programação. Os recursos criados podem ser salvos no servidor.
GLO Maker	Livre para <i>download</i> . Possibilita a criação de OA em formato de slides que podem ser adaptados. Faz uso de templates flexíveis. Após desenvolvidos, podem ser executados na <i>web</i> ou em um AVA.
LOC	<i>Open Source</i> . Possibilita a criação de recursos em formato de slides, utilizando <i>templates</i> flexíveis. Após desenvolvido, o recurso pode ser salvo em formato ZIP e armazenado em um AVA.
RELOAD	Livre para <i>download</i> . Possibilita criar, compartilhar e reutilizar OA, além de gerar plano de aula para os mesmos. Os recursos são criados em formato de slides. Uma vez desenvolvidos são empacotados, compactados e compartilhados em um AVA. Faz uso dos padrões Scorm e IMS.
XICAL	<i>Open Source</i> . Possibilita a criação de slides em formato <i>media player</i>
EXE Learning	<i>Open Source</i> . Possibilita a criação de conteúdos educacionais em páginas web . Faz uso dos padrões IMS e SCORM.
PACHYDERM	Livre para <i>download</i> . Possibilita a composição de recursos multimídia, gerando recursos interativos em formato de <i>player</i> de vídeo e áudio. De Fácil utilização. Possui <i>templates</i> para auxiliar na criação. Os recursos criados podem, caso queiram, ser armazenados no repositório que leva o mesmo nome da ferramenta (Pachyderm)

Fonte: Própria Autora

*players*. Entretanto, as ferramentas apresentadas não contemplam os problemas inerentes à multidisciplinaridade da equipe e o tipo de recurso produzido é limitado, implicando assim na redução da compreensão dos conceitos abordados em sala de aula pelo professor.

## 2.3 Discussão

Os diversos modelos empregados na produção de recursos educacionais associados às técnicas de adaptação, são apresentados neste capítulo como o cerne do desenvolvimento e adaptação de RED. Dessa forma, é importante levantar a discussão se os modelos apresentados atendem, de fato, às peculiaridades (equipe multidisciplinar e documentação utilizada no processo) inerentes à produção deste tipo de recurso e se as ferramentas de autoria são capazes de dar o suporte necessário aos professores e educadores de modo a suprimir as dificuldades técnicas referentes a adaptação dos recursos desenvolvidos.

Em relação aos processos empregados na produção de objetos de aprendizagem, nenhum deles tem de fato, inserido em seu contexto, a preocupação de minimizar as limitações enfrentadas pela equipe multidisciplinar quando o assunto se refere à clara compreensão da documentação produzida e utilizada durante o desenvolvimento do recurso. Diante dessa situação, questionamentos sobre a necessidade de se desenvolver diversos processos para a produção de OA podem ser levantados, haja vista que nenhum dos processos descritos evidencia a preocupação em minimizar as problemáticas existentes na comunicação dos membros da equipe multidisciplinar. Além disso, alguns desses processos fazem uso de metodologias ágeis como forma de dinamizar a produção dos recursos, quando na verdade o problema principal na produção desses recursos não está relacionado ao tempo destinado para desenvolvê-lo, mas sim uma correta compreensão do que precisa ser especificado em seu projeto. Se o tempo de desenvolvimento for reduzido e nada for feito com relação ao tipo de documentação produzida e utilizada, as problemáticas persistirão, pois apenas se terá de modo mais rápido um recurso com prováveis erros de especificação.

Já com relação às ferramentas de autoria, é possível ressaltar que mesmo frente as limitações proporcionadas pelos processos de desenvolvimento existentes em desenvolver tais recursos educacionais, a grande importância e contribuição que estas ferramentas fornecem, tanto para quem produz quanto para quem utiliza esses materiais, são evidentes. Isto porque no que se refere ao desenvolvimento, a figura da equipe multidisciplinar é suprimida, deixando em seu lugar a figura de um único indivíduo interessado

em desenvolver o recurso, que na maioria das vezes é o professor, permitindo, assim, que os mesmos desenvolvessem seus próprios recursos a partir de suas necessidades sem se depararem com obstáculos referentes à programação e/ou design.

Muito embora os ganhos proporcionados pelas ferramentas de autoria sejam inquestionáveis, ainda assim elas apresentam limitações quanto ao tipo do recurso produzido, isto porque um número considerável de ferramentas desenvolvem recursos no formato de slides ou *media players*, mas quando se referem a recursos que envolvam uma maior interação do usuário, elas deixam a desejar. Outro aspecto importante e que não é contemplado por tais ferramentas é o desenvolvimento de recursos passíveis de serem adaptados, pois realizar qualquer alteração no recurso produzido por tais ferramentas requer que o mesmo seja desenvolvido novamente, fazendo com que se recorra a duas falhas, sendo a primeira relacionada a não reutilização e a segunda ao consequente gasto de tempo em uma nova produção.

## 2.4 Conclusão

Como apresentado neste capítulo, existem diversos processos de desenvolvimento destinados à produção de RED. Entretanto, iniciativas nacionais e internacionais preocupam-se, basicamente, na análise, projeto e desenvolvimento de recursos monolíticos, que impossibilitam, uma vez desenvolvidos, a realização de qualquer alteração ou intervenção na interface do mesmo por parte dos professores, isto porque toda e qualquer alteração só é possível ser realizada com a alteração de código.

Outro problema não contemplado pelos processos descritos é o tipo de documentação gerada e utilizada pelos membros da equipe multidisciplinar. As documentações existentes em todos os processos analisados não fazem alusão a nenhum mecanismo formal que possa ser compreendido por todos os membros da equipe, acarretando assim uso de linguagem informal implicando em possíveis atrasos decorrentes das interpretações dos requisitos especificados nas mesmas.

É importante ressaltar que não basta apenas produzir RED, é necessário observar que o mesmo precisa ser bem especificado para poder viabilizar a otimização com

---

relação ao tempo de desenvolvimento e possibilitar que o recurso atinja a qualidade desejada.

A produção de recursos monolíticos pode ser também observada através dos RED criados a partir das ferramentas de autoria, que permitem criar quantos RED sejam necessários, porém para realizar alterações, novos recursos devem ser criados.

É bem verdade que em meio à diversidade de processos e ferramentas apresentadas, os professores têm a liberdade para criar os seus materiais de modo a atender suas reais necessidades. No entanto, vale ressaltar que o obstáculo no desenvolvimento e utilização dos RED ainda persiste, dado que os recursos desenvolvidos com o auxílio dessas ferramentas não são susceptíveis de serem facilmente adaptados ou customizados.

## Capítulo 3

### Customização Guiada

“Mudar é difícil mas é possível.”  
– Paulo Freire

Neste capítulo é apresentada a estratégia proposta nesta tese de doutorado para a adaptação de recursos educacionais digitais (RED), denominada de customização guiada. Para tanto, são apresentadas nas seções a seguir, o significado, a importância e o nível de contribuição que essa estratégia pode oferecer ao desenvolvimento dos RED bem como sua relevância ao contexto educacional.

#### 3.1 Justificativas e Conceito

Inúmeros são os recursos desenvolvidos e disponibilizados em repositórios para serem utilizados no âmbito educacional por professores e educadores (AMIEL; SQUIRES; OREY, 2009). No entanto, a forma como esses recursos são desenvolvidos não contemplam a necessidade que o professor tem de poder gerenciar suas decisões com relação ao sequenciamento e representação do conteúdo abordado no mesmo (SOUZA; CASTRO-FILHO; ANDRADE, 2011a). Essa dificuldade no gerenciamento ocorre porque os recursos desenvolvidos possuem uma estrutura conceitual e de representação fechadas, na qual toda e qualquer alteração só pode ser realizada no código fonte (SOUZA; CASTRO-FILHO; ANDRADE, 2010b). No entanto, essa ação de gerenciamento, como apresentada no Capítulo 2, é imprescindível para o professor construir e aplicar sua autonomia docente. Nesse contexto, a seguinte questão pode ser levantada: como permitir que sejam realizadas mudanças em um RED por indivíduos que não possuem habilidades para modificar o código de forma a possibilitar a adaptação desse recurso às suas reais necessidades?



Esse problema ocorre porque, conforme descrito no Capítulo 2, os processos de desenvolvimento empregados nesses recursos acabam por gerar produtos monolíticos nos quais, praticamente a única forma de realizar qualquer adaptação é através da manipulação do seu código fonte. Contudo, na grande maioria das vezes, nem mesmo esses códigos estão disponíveis. Esse fato restringe severamente a autonomia do professor com relação à utilização de um RED como elemento de suporte às suas aulas, visto que os conteúdos é que acabam sendo adaptados às características dos recursos disponíveis.

Essa restrição à autonomia do professor dentro do processo de planejamento de utilização do recurso se dá pelo fato de não haver mecanismos diretos que possibilitem qualquer tipo de alteração no mesmo. Essa restrição não se aplica somente à forma como o conteúdo é apresentado, mas também aos tipos de recursos a serem utilizados, uma vez que fatores como idioma e modo de representação do conceito não deveriam inviabilizar a utilização de um RED em qualquer que fosse a sala de aula (AMIEL; SQUIRES; OREY, 2009).

Segundo Dagiene e Zilinskiene (2009), a possibilidade de modificar um recurso ou mesmo compartilhá-lo dentro ou fora de seu país vem se tornando uma questão chave da política educacional europeia, isso porque o desenvolvimento constante das Tecnologias da Informação e Comunicação (TIC) tem desencadeado novas necessidades na educação. Necessidades essas que requerem intervenções diretas por parte do professor, tanto a nível contextual quanto cultural em um RED, como forma de facilitar a assimilação do conceito ou conteúdo em estudo.

Em meio a essa necessidade, é proposta nesta tese uma nova estratégia de adaptação a ser empregada no desenvolvimento de RED, a qual foi denominada de Customização Guiada (SOUZA; CASTRO-FILHO; ANDRADE, 2011b).

Dessa forma, a Customização Guiada baseia-se no conceito de customização apresentado em Freund (2008) e é definida nesta tese da seguinte forma:

---

**Customização Guiada é uma estratégia de adaptação que tem por objetivo possibilitar que os próprios clientes façam reconfigurações ou ajustes diretamente na interface de um RED no intuito de adequá-lo a um novo contexto ou a novas necessidades.**

---

Recursos desenvolvidos sob a perspectiva desta estratégia são denominados RED customizáveis. É importante ressaltar, que o termo “guiada” diz respeito à necessidade de se controlar a realização das adaptações por parte dos clientes, visto que, pelo fato dos RED serem recursos que incorporam diversas relações pedagógicas na sua concepção, essas não podem ser perdidas em virtude de adaptações não controladas.

Desta forma, a customização guiada é uma atividade que deve ser realizada pelo cliente quando da utilização do recurso, isto porque seu objetivo é desvincular a ideia de que o planejamento do professor é que deve se adequar para utilizar um recurso, mas tornar também possível a adequação do recurso ao planejamento do professor (cliente), proporcionando assim, uma maior autonomia docente.

Para uma melhor compreensão, são apresentados na Figura 3.1, em quais etapas do processo de desenvolvimento de um RED customizável a estratégia da customização guiada deve atuar. Nas subseções seguintes essas etapas são detalhadas.

Figura 3.1: Customização Guiada no Contexto de Desenvolvimento de um RED



Fonte: Própria Autora

### 3.1.1 Concepção

A etapa de concepção dentro do processo de desenvolvimento de um RED abrange desde a ideia motivacional até a descrição de cada cenário do recurso. É na descrição dos cenários que os profissionais que atuam nessa fase sinalizam o que pode ser modificado e o tipo de modificação possível a ser realizada.

Cada cenário descrito é formado por elementos e ações que possuem, em sua maioria, potencial pedagógico (PIVELLI, 2006). Como exemplo de elementos que podem fazer parte de um RED é possível citar: textos, imagens, botões, vídeos e áudios. Sendo assim, o potencial pedagógico é definido nesta tese da seguinte forma:

---

**A capacidade que um elemento ou ação tem, isoladamente ou em conjunto com outros elementos, de expressar ou representar um determinado conceito, favorecendo assim a sua compreensão.**

---

Esses elementos podem possuir potencial pedagógico, porque através deles pode ser possível fazer a descrição ou representação de um conceito (GAUTHIER; TARDIF, 2010) ou mesmo estabelecer uma interação com o usuário do RED. Por exemplo, um botão na interface de um RED que oferece acesso a uma atividade específica do recurso possui potencial pedagógico, visto que a utilização desse elemento leva à exploração de um conceito associado. É importante ressaltar que elementos podem ser combinados na interface de um RED, como estratégia de favorecimento ao aprendizado do usuário. Daí a importância de dedicar uma maior atenção à etapa de concepção. Mais adiante, na Seção 3.3, o conceito de potencial pedagógico é melhor explorado.

Ainda no momento da concepção devem ser criados documentos que descrevam os cenários e que auxiliem a equipe de desenvolvimento a produzir as interfaces do RED. Esses documentos, que serão denominados de forma genérica de roteiros, mas que dependendo do tipo de RED podem ser *storyboards*, *wireframes*, entre outros, servem para representar os elementos e ações que irão compor o recurso. São exatamente esses ou alguns desses elementos e ações que podem ser indicados para serem customizados. O que irá determinar se um elemento ou ação pode ser customizado é a capacidade que

ele tem, segundo a equipe que o idealizou na etapa de concepção, de uma vez modificado, não acarretar alterações nos objetivos pedagógicos do recurso. É válido observar que na concepção devem participar profissionais que tenham a competência pedagógica necessária para realizar as devidas indicações nos elementos a serem customizados.

Uma vez identificado os elementos e ações passíveis de sofrerem alguma alteração, o recurso entra na etapa do processo, denominada na Figura 3.1, de desenvolvimento. Nesta etapa, deve ser implementado o que foi delineado no roteiro, incluindo também as ações relativas aos atributos de customização dos elementos e ações previamente sinalizadas na etapa de concepção. Maiores detalhes sobre esta etapa são abordados no Capítulo 4 desta tese.

Diante dessa possibilidade, serão apresentadas a seguir, as implicações da customização guiada na utilização de um RED customizado.

### 3.1.2 Utilização

Ao se utilizar um RED customizável, tem-se a possibilidade de ajustar o recurso ao planejamento de uma aula e sua aplicação ao contexto educacional.

A vantagem em utilizar um RED customizável é que o cliente, no caso o professor, tem autonomia para determinar, por exemplo, a sequência na qual as informações devem ser apresentadas e por conseguinte estruturar melhor sua estratégia didática.

Essa estruturação ocorrerá por intermédio das ações a serem realizadas nos elementos customizáveis diretamente na interface do recurso, deixando de lado a barreira na qual só seriam permitidas a realização de modificações ou adaptações diretamente nos códigos.

Conforme apresentado no Capítulo 1, o indivíduo só tem autonomia se ele tiver liberdade para executar, simultaneamente, o pensar e fazer autônomo ou seja, o sujeito deve ser capaz de resolver questões por si mesmo, de tomar decisões sempre de maneira consciente e arcar com as consequências de seus atos. (FREIRE, 1996).

Nesse contexto, pode ser observado a introdução de um determinado grau de abertura através dos RED customizáveis. Essa abertura está relacionada, como apresentado

na Seção 3.2, com o que é defendido pelos Recursos Educacionais Abertos (REA).

## 3.2 A Relação entre RED Customizável e REA

Na era da globalização, o termo compartilhar deixou de ser uma opção e passou a ser uma necessidade, principalmente quando esse compartilhamento faz referência à informação e ao conhecimento (SOUZA; CASTRO-FILHO; ANDRADE, 2011a).

Nesse contexto de abertura e colaboração em prol do compartilhamento de informações e conhecimento, surgiu o movimento em defesa da abertura dos Recursos Educacionais Digitais (RED), denominado Recursos Educacionais Abertos (REA) ou *Open Educational Resource* (OER).

Segundo Dutra e Tarouco (2011), a filosofia empregada pelo movimento REA pode ser um importante instrumento para a disseminação e universalização do conhecimento.

Segundo Hylén (2006), o REA é um fenômeno relativamente novo, que pode ser visto como parte de uma tendência maior para a abertura em ensino superior, incluindo movimentos mais conhecidos mundialmente e estabelecidos, como *Software Open Source* (OSS) (INITIATIVE, 2007) e *Open Access* (OA) (SUBER, 2004; SUBER, 2009).

Desse modo, com novas tecnologias, diversos aspectos de abertura estão associados aos recursos educacionais, tais como conteúdo aberto, recursos abertos, ensino aberto, arquiteturas abertas no design de espaços físicos e virtuais, código aberto, e conhecimento aberto (BATSON; PAHARIA; KUMAR, 2008).

De acordo com Hylén (2006), essa abertura deve ocorrer tanto em nível de disponibilização quanto de utilização do recurso, não devendo haver barreiras técnicas, financeiras (custo) e legais (licenças) para o usuário final, pois somente assim ele será capaz não só de usar ou ler o recurso, mas também de adaptá-lo e conseqüentemente reutilizá-lo.

Essa possibilidade de modificar um recurso ou mesmo compartilhá-lo dentro ou fora de seu país vem se tornando uma questão chave da política educacional, isso

porque, como apresentado anteriormente, o desenvolvimento constante da tecnologia da informação e comunicação (TIC) tem desencadeado novas necessidades na educação (DAGIENE; ZILINSKIENE, 2009).

Sendo assim, a ideia de desenvolvimento e utilização de RED customizável, proposta nesta tese, tem por objetivo, contribuir com o movimento de abertura e democratização do conhecimento no âmbito educacional. Contudo, o conceito de customização guiada amplia a aplicabilidade dessa abertura, visto que os recursos poderão também ser modificados por clientes com pouco ou nenhum conhecimento de programação.

No entanto, ao se pensar no desenvolvimento de um RED, é importante detectar os possíveis elementos que compõem a sua interface e o potencial pedagógico de customização de cada elemento.

Dessa forma, é apresentado na Seção 3.3, o modo de se identificar, na fase de concepção, o potencial pedagógico dos elementos que compõem a interface de um recurso, além dos tipos de permissões concedidas para realizar as customizações desses elementos.

### 3.3 Potencial Pedagógico

Conforme apresentado no Capítulo 2, podem ser classificados como RED os softwares educativos, objetos de aprendizagem (OA), imagem, áudio, vídeo e hipertexto. Os do tipo OA, por exemplo, buscam conter em sua interface, na maioria das vezes, elementos que sirvam de mecanismos favorecedores da contextualização do conceito tratado. Exemplo disso, são os RED do tipo imagem, áudio e vídeo que podem ser utilizados tanto individualmente quanto inseridos dentro de um OA, como forma de ampliar a capacidade de compreensão do conceito abordado pelo recurso.

Quando inseridos no recurso, independente do formato (imagens, textos, áudio, vídeo, dentre outros) assumido, é atribuído a cada um dos elementos um potencial pedagógico que agregado aos outros elementos de sua interface, tem o propósito de favorecer o aprendizado do usuário.

Para uma melhor compreensão da importância do conceito de potencial pedagógico, é apresentado na Figura 3.2 a tela de recurso educacional SumaMonedas que pode ser localizado no banco internacional de objetos educacionais(BIOE)<sup>1</sup>. Este recurso trabalha com operações aritméticas, mas especificamente adição. Levando em consideração que as operações aritméticas são trabalhadas nas séries iniciais do ensino fundamental, esse conteúdo pode ser empregado em qualquer sala de aula. No entanto, aspectos culturais, econômicos e também educacionais devem ser observados. Um exemplo disso é o idioma (espanhol) empregado no recurso, que pode ser um obstáculo quando o mesmo for utilizado por usuários que não têm compreensão do mesmo. Dessa forma, circunstâncias como essa, podem acarretar desvio na atenção do usuário do foco principal do recurso, que no caso deveria ser a compreensão do conceito matemático abordado, para a compreensão do idioma.

No entanto, para que a utilização do recurso apresentado na Figura 3.2, seja viável para um usuário de qualquer nacionalidade, tendo em vista que o idioma não deva ser qualificado como pré-requisito, é importante possibilitar que alterações em nível textual sejam realizadas, de modo que as informações apresentadas em sua interface, possam ser traduzidas para a língua nativa do usuário. Neste caso, pode-se afirmar que o elemento texto deste recurso possui potencial pedagógico.

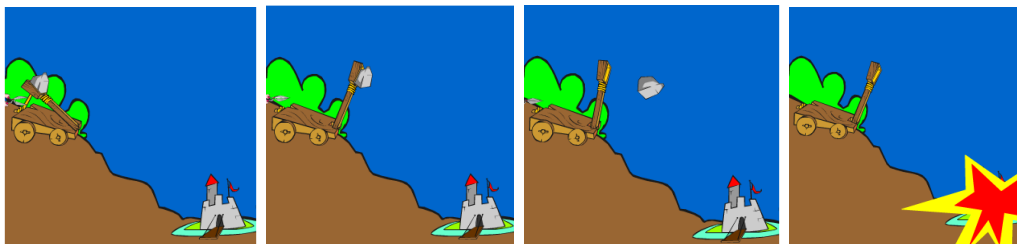
Figura 3.2: Pontencial Pedagógico no SumaMonedas Vedoque



Fonte: <http://www.vedoque.com/juegos/suma-monedas.swf>

<sup>1</sup> <http://www.vedoque.com/juegos/suma-monedas.swf>

Figura 3.3: RED que Aborda a Compreensão e a aplicabilidade da Função Quadrática



Fonte: <http://rived.mec.gov.br/modulos/matematica/funcoes/atividade1.htm>

É importante ressaltar a existência de elementos em um RED que não possuem potencial pedagógico. Um exemplo disso pode ser observado na Figura 3.2, pois as imagens dos personagens que estão logo abaixo do quantitativo de moedas são utilizadas apenas para ajudar na composição da interface do recurso, sua presença ou ausência não interfere na compreensão do conceito tratado pelo mesmo.

Outro aspecto que deve ser levado em consideração são RED que nenhum dos seus elementos possui potencial pedagógico. Isso não significa dizer que o recurso tenha qualidade pedagógica inferior a outro recurso que apresente elementos com esses potenciais. Um exemplo disso é o RED apresentado na Figura 3.3.

Com o RED da Figura 3.3 é possível abordar à aplicação do conceito de função quadrática, como por exemplo o gráfico gerado por essa função a partir do lançamento de uma pedra pela catapulta. No entanto, as variáveis necessárias que são utilizadas para gerar o gráfico não estão presentes no recurso, pelo menos não de forma explícita em sua interface. Outro ponto importante para ser analisado, é que independente da ferramenta utilizada para lançar a pedra, ela continuará sendo uma representação do conceito tratado inicialmente pelo recurso, no caso função quadrática.

Dessa forma, é possível observar que o recurso tem por objetivo demonstrar somente a aplicabilidade do conceito. RED dessa natureza não são passíveis de serem customizados, através da estratégia da customização guiada.

Já os RED que têm condições de serem customizáveis são aqueles que possuem em sua interface elementos que apresentam potencial pedagógico. Sendo assim, é importante salientar, como já mencionado no início desse capítulo, que a definição do que deve ou não ser alterado é sinalizado ainda na fase de concepção com a intervenção



de um profissional que tenha conhecimento para identificar o potencial pedagógico dos elementos disponibilizados nos RED. Essa indicação sobre a intervenção a ser realizada guiará o professor a realizar a customização necessária no recurso.

Para exemplificar, a Figura 3.4 apresenta um RED com os elementos imagem, texto, áudio e botão.

Figura 3.4: RED Fazenda Rived



Fonte: [http://rived.mec.gov.br/atividades/matematica/fazenda/mat1\\_ativ1.swf](http://rived.mec.gov.br/atividades/matematica/fazenda/mat1_ativ1.swf)

Esse recurso, denominado de Fazenda Rived<sup>2</sup> tem por objetivo auxiliar na compreensão de conceitos relacionados à definição de conjuntos e suas relações de pertinência. Esse recurso, diferente do recurso apresentado na Figura 3.3, possui elementos com potencial pedagógico como, por exemplo, o texto, representado pela fala do personagem (fazendeiro) juntamente com o áudio que reproduz o texto escrito presente no mesmo. Dessa forma, se o conceito de customização guiada fosse aplicado ao projeto de arquitetura desse recurso, mas especificamente aos elementos de texto e áudio, resultaria na geração do Fazenda Rived customizável.

Como qualquer intervenção nos elementos pode afetar, com maior ou menor intensidade, as características pedagógicas do recurso, é importante delimitar ainda na fase de concepção de que modo serão realizadas as intervenções de modo a preservar o interesse e o conhecimento do professor para realizar as devidas adaptações. Deste modo,

<sup>2</sup> [http://rived.mec.gov.br/atividades/matematica/fazenda/mat1\\_ativ1.swf](http://rived.mec.gov.br/atividades/matematica/fazenda/mat1_ativ1.swf)

foi introduzido o conceito de Grau de Liberdade apresentado na seção 3.4. para evitar que possíveis falhas acarretadas pelas intervenções possam afetar o conceito abordado no recurso.

## 3.4 Grau de Liberdade

No intuito de valorizar o conhecimento dos clientes e viabilizar as intervenções nos RED customizáveis, é proposto neste trabalho o conceito de Grau de Liberdade (GL). O GL pode ser definido da seguinte forma:

---

**Grau de Liberdade é um conjunto de permissões associadas a um cliente para modificar um RED levando em consideração o seu nível de conhecimento para realizar tal modificação.**

---

O GL tem por objetivo permitir que adaptações sejam realizadas no recurso levando em consideração os elementos com potencial pedagógico presente no mesmo e o nível de conhecimento requerido do cliente para realizá-las.

A liberdade atribuída ao cliente para realizar as intervenções é baseada na ideia de liberdade defendida pelo movimento do software livre (STALLMAN et al., 1996) que indica, sem restringir, o tipo de modificação possível a ser realizada pelo indivíduo em um software, dentro de uma escala de 1 a 4 . Nesta tese, cada GL possibilita que o cliente realize modificações que vão desde alterações em alto nível (Grau 1), como por exemplo, modificações simples na interface do recurso, bem como as de baixo nível, como por exemplo, modificações no código fonte do recurso (Grau 4).

No grau 1, ou simplesmente GL 1, o cliente tem condições de realizar as ações de customizações do tipo desabilitar ou habilitar além de esconder ou mostrar elementos da interface que foram indicados na concepção do recurso.

Já no GL 2, os clientes herdam as ações de customização do GL 1 adicionando a possibilidade de realizar a fragmentação de um vídeo, no qual será possível delimitar o limite inferior e superior de trechos do vídeo a ser apresentado ao usuário final do recurso.

Com relação ao GL 3, além das prerrogativas de clientes de GL 1 e 2, é possível a realização de edições de elementos textuais e substituições de elementos.

Os GL 1, 2 e 3, geralmente são destinados aos indivíduos que não desejam ou não têm conhecimento para realizar alterações em baixo nível (código).

Por fim, o GL 4 pode ser considerado o nível mais alto de abertura a ser alcançado por um indivíduo em um RED customizável, pois ele herda as permissões oriundas dos GL 1, 2 e 3, e adiciona as ações de intervenção em nível de código. Nesse GL, a estratégia da customização guiada é parcialmente aplicada, haja vista que o acesso ao código fonte do recurso possibilita que qualquer tipo de intervenção seja feito no recurso, mas para realizá-la é necessário que o cliente tenha conhecimento de programação. Podem ser considerados exemplos de modificações exclusivas a serem realizadas nesse nível (GL 4), a alteração ou inclusão de *feedbacks*, pois qualquer tipo de intervenção nesse elemento requer um conhecimento mais especializado, porque adaptá-lo, apesar de ser um elemento textual, equivale a modificar trechos do código do recurso.

É importante ressaltar que o cliente só terá condições de realizar as adaptações nos elementos que compõem a interface de um RED, se for concedida, ainda na fase de concepção, a permissão necessária para alterá-los. Dessa forma, quem projeta o recurso é responsável por determinar se o elemento pode sofrer alguma adaptação e, caso possa, determinar o tipo de intervenção permitida.

O GL busca a associação do “querer modificar” com o “poder modificar”, no que diz respeito à utilização e adequação dos RED por todo e qualquer indivíduo. Dessa forma, o objetivo maior dessa liberdade é viabilizar além da abertura do conteúdo, a abertura do recurso, possibilitando assim o compartilhamento não só de ideias, mas também do conhecimento.

Como pode ser observado, existe uma relação direta entre GL e Nível de Customização de um RED customizável. O nível de customização está relacionado ao recurso. Ele indica se os elementos de um RED têm condições de serem modificados. Já o GL indica o nível de conhecimento necessário que o cliente deve ter para poder realizar essas alterações. Por exemplo, um recurso pode possuir elementos do tipo imagem, texto, áudio e vídeo. No entanto quem projetou o recurso só permitiu que intervenções

de desabilitar e esconder fossem realizadas. Dessa forma, mesmo que o professor tenha GL que possibilite realizar a fragmentação do vídeo, ele não poderá fazê-lo adotando a estratégia da customização guiada, haja visto que tal permissão não foi concedida a esses elementos. Esse cuidado de restringir o que pode ou não ser customizado visa preservar tanto o autor quanto o objetivo pedagógico a ser alcançado no recurso.

### 3.5 Discussão

Com base na política de compartilhamento defendida pelo movimento dos Recursos Educacionais Abertos (REA), surgem ideias que remetem aos mais diversos tipos de aberturas para favorecer aqueles que fazem uso desse tipo de recurso (AMIÉL; OREY; WEST, 2010).

Diante desse cenário, e influenciados pela globalização das informações nos mais diversos setores da sociedade (econômico, social e cultural), é possível acreditar que não adianta somente criar conteúdos ou recursos, é necessário estabelecer relações mais próximas das novas tendências mundiais. Tendências essas que caminham não só para o compartilhamento de ideias e conhecimento, mas também para uma reutilização sistematizada nos moldes da engenharia de software (FRAKES; KANG, 2005).

Sendo assim, a estratégia de adaptação de RED proposta nesta tese de doutorado denominada da customização guiada, surge como mais um elemento para reforçar a ideia de reutilização e o movimento dos REA, pois sua aplicação suscita a condição de desenvolver recursos já capazes de serem modificados no momento em que os professores estão planejando utilizá-los em sala de aula. Isto porque a customização guiada não foca somente na produção, mas também na utilização do recurso.

No entanto, essas modificações permitem levantar a discussão sobre o tipo de alteração possível. No caso desta tese, foi utilizado o conceito de potencial pedagógico e grau de liberdade para reforçar a estratégia proposta. Sendo que o primeiro diz respeito a quem irá desenvolvê-lo e o segundo a quem irá utilizá-lo.

Outra questão importante é o tipo de ferramental disponível para auxiliar nesse processo de adaptação. Com base nisso, é importante pensar em ferramentas que

tornem a abordagem da customização guiada algo possível e sistemático, para que os RED desenvolvidos sob essa ótica possibilitem aos clientes a inserção de suas ideias e conhecimentos. Para isso, é necessário buscar meios que reduzam o tempo destinado a implementar e centre sua atenção em como executar.

## 3.6 Conclusão

Neste capítulo é proposto o conceito de customização guiada como uma nova estratégia a ser empregada ao desenvolvimento e utilização de recursos educacionais digitais (RED). Essa abordagem possibilita elevar o nível de abertura do recurso e a consequente valorização da autonomia docente em seu planejamento, pois há uma preocupação em buscar o equilíbrio entre o desenvolvimento e a utilização do recurso.

Além da busca pelo equilíbrio, o aspecto da reutilização deve ser considerado quando a questão é o desenvolvimento de RED. No entanto, este aspecto deve ser viabilizado sempre que houver necessidade de adequação do recurso a novos contextos e não apenas quando se tem acesso ao código fonte do recurso.

Diante da preocupação em aproximar o desenvolvimento do recurso a sua utilização, acreditamos que a abordagem da customização guiada é um passo adiante no sentido de possibilitar que conteúdos e recursos possam ser igualmente compartilhados.

Assim, como forma de demonstrar a aplicabilidade desse novo conceito, é apresentado no Capítulo 4 como utilizar a estratégia da Customização Guiada no desenvolvimento e utilização de um tipo específico de RED, denominado de Objeto de Aprendizagem.

## Capítulo 4

# CLAssRoOM no Desenvolvimento de Objetos de Aprendizagem Customizáveis

“Ninguém é sujeito da autonomia de ninguém.”

– *Paulo Freire*

A produção de OA vem acontecendo desde meados da década de 90 e já existe um número significativo desses recursos disponíveis atualmente. Entretanto, a produção e sua conseqüente reutilização, não tem acontecido de maneira satisfatória devido a diversas limitações (WILEY, 2010). Desta forma, o presente capítulo apresenta um processo de produção, denominado CLAssRoOM, que utiliza a estratégia de customização guiada proposta no Capítulo 3, associada à estratégia de engenharia de software orientada a modelos. É importante ressaltar que o referido processo desenvolve um tipo específico de OA adaptável, denominado de Objeto de Aprendizagem Customizável (OAC). Dessa forma, serão apresentados o conceito e a importância de um OAC para o contexto educacional bem como a sistemática de funcionamento do processo CLAssRoOM na produção deste tipo de recurso.

### 4.1 Objetos de Aprendizagem Customizáveis

Como forma de demonstrar a aplicação da proposta de Customização Guiada, neste capítulo esse conceito é utilizado no contexto dos Objetos de Aprendizagem, produzindo um tipo de OA adaptável, denominada de Objetos de Aprendizagem Customizáveis (OAC). Nesta tese, são utilizados OA do tipo animação e simulação na implementação dos OAC. A escolha por este tipo específico de OA pode ser justificada pelo fato dos OA se destacarem como RED capazes de serem reutilizados nos mais diferentes contextos e,

portanto, são fonte de intensa pesquisa nas últimas décadas (OCHOA; DUVAL, 2008). Além disso, esses recursos possibilitam um maior grau de interação com os alunos, acarretando ganhos ao contexto educacional.(NICOLEIT et al., 2010).

Na literatura há evidências de OA personalizados (PALAZZO, 2002). Embora sejam adaptados, estes OA diferem dos OAC, conforme descrito no Capítulo 2, pela forma como as modificações são realizadas no recurso, pois enquanto a personalização requer pessoas com conhecimento específico para realizar as modificações, a customização permite que qualquer professor possa realizá-las diretamente na interface do recurso. Para uma melhor compreensão desse conceito, a subseção 4.1.1 apresenta a definição de OAC.

### 4.1.1 Definição de OAC

Os OAC são definidos nesta tese como

---

**Objetos de Aprendizagem que possibilitam realizar intervenções guiadas diretamente em sua interface.**

---

Essas intervenções podem ser consideradas uma forma de abertura e têm por objetivo suscitar a capacidade de reutilização dos recursos e assim favorecer a adaptação dos mesmos a diversas formas e contextos educacionais.

É importante ressaltar que um OAC é um OA com permissões de customização atreladas aos elementos que compõem sua interface. A vantagem dos OAC sobre os OA ou qualquer outro recurso empregado no contexto educacional, é que a estratégia de customização guiada atrelada ao seu desenvolvimento permite que o professor seja sujeito da produção do saber (FREIRE, 1996), através das intervenções que ele pode realizar na interface do mesmo, buscando, assim, encontrar o equilíbrio entre o conceito, o contexto e sua prática docente.

Nesse sentido, será apresentado na subseção 4.1.2 algumas das razões para se realizar customização neste tipo de recurso.

### 4.1.2 Por que customizar

Apesar do grande número de OA disponíveis na *web*, o número de educadores que usam este tipo de material em sala de aula ainda é baixo. Isso acontece devido ao fato de não se sentirem a vontade em fazer uso de tais recursos (SOUZA et al., 2007).

De acordo com NIELSEN (2009), os professores necessitam de ferramentas para desenvolver a sua autonomia na criação de ambientes que fascinem e motivem os alunos. Esta independência para desenvolver tais ambientes ajudará os professores a criar diferentes oportunidades de aprendizagem.

Assim, acredita-se que a utilização de um OAC pelo professor, além de permitir ganhos educacionais em sala de aula, pode oferecer vantagens financeiras, por causa da menor quantidade de tempo gasto, para fazer pequenas adaptações ou ajustes nos recursos.

Outra motivação para produzir e disponibilizar OAC, por exemplo, seria a possibilidade de viabilizar o tratamento de aspectos culturais neste recursos, fazendo com que os professores possam contribuir de modo mais efetivo com o conhecimento dos alunos, levando em consideração o nível de diversidade cultural dentro e fora de sua comunidade. Esta diversidade pode ser encontrada, por exemplo, ao se utilizar frutas para explicar um conceito em um OA. Exemplo disso é a utilização da fruta coco em um recurso, que é típica de países tropicais, podendo ser substituída por uma fruta diferente, como morango, caso os alunos não estejam familiarizados com a primeira fruta.

É bem verdade, que nem sempre a customização do recurso será necessária, prova disso é o recurso da Figura 3.3 apresentado no Capítulo 3, que serve para mostrar a aplicabilidade do conceito, mas é de fundamental importância que os professores tenham condições de discernir sobre o que utilizar e de que modo utilizar.

Diante da importância de se customizar um recurso, é necessário compreender o que pode ser customizado. Dessa forma, é apresentado na subseção 4.1.3 o que deve ser customizado em um OA e os tipos de customizações possíveis de serem realizadas.



Quadro 4.1: Tipos de Elementos e Intervenções Adaptativas em um OAC

ELEMENTOS	AÇÕES DE CUSTOMIZAÇÃO				
	HABILITAR/DESABILITAR	ESCONDER/MOSTRAR	EDITAR	SUBSTITUIR	FRAGMENTAR
Imagem	X	X		X	
Botão	X	X			
Texto	X	X	X		
Áudio	X	X		X	
Vídeo	X	X		X	X

Fonte: Própria Autora

### 4.1.3 O que customizar

Diante da possibilidade de customizar um OAC, é importante atentar o que customizar em um recurso para que os conceitos tratados no mesmo, sejam preservados, pois a importância de customizar um OAC não está relacionada diretamente à obtenção de um outro recurso, mas sim à capacidade de criar novas estratégias de ensino a partir das possíveis reconfigurações na interface do mesmo.

Desse modo, são apresentados no Quadro 4.1, alguns dos elementos que podem compor a interface de um OAC juntamente com as possíveis ações de customização que cada um desses elementos podem ser submetidos.

Conforme apresentado no Quadro 4.1, as ações de customização passíveis de serem realizadas nos elementos que compõem a interface de um OAC, são do tipo: desabilitar/habilitar, esconder/mostrar, editar, substituir e fragmentar. Para simplificar a citação dessas ações ao longo do texto, daqui por diante serão utilizados apenas os seguintes termos: desabilitar, esconder, editar, substituir e fragmentar.

A ação de **desabilitar** tem por finalidade restringir o acesso do usuário, no caso o aluno, a um espaço pré-determinado, como por exemplo, outros cenários, situações, ou níveis de atividades. A vantagem em se fazer uso dessa ação é deixar o aluno ciente de

que outros espaços podem ser explorados e acessados, mas os mesmos serão habilitados gradativamente, à medida que o professor necessitar aprofundar o conceito abordado. Isso porque, uma vez o elemento desabilitado, o usuário poderá visualizá-lo, mas não poderá acessá-lo. Exemplos de elementos que podem sofrer esse tipo de ação: imagem, botão, texto, áudio e vídeo.

Assim como a ação de **desabilitar**, a ação de **esconder** um elemento tem por objetivo restringir o acesso ao aluno a um espaço pré-determinado. A diferença com relação à ação de desabilitar está relacionada ao interesse do professor, pois enquanto desabilitar permite que o aluno tenha conhecimento do que está por vir, o esconder retira totalmente do campo de visão do usuário as outras atividades, cenários ou mesmo situações que o recurso possa oferecer e que não dizem respeito, tanto em conceito quanto em nível de complexidade, com o planejamento do professor.

A ação de **editar** está relacionada somente ao elemento texto. Esta, por sua vez, pode ser realizada sempre que houver necessidade de reformulá-lo ou simplesmente ajustá-lo ao público-alvo. Exemplos disso são: falas dos personagens, textos com instruções, *feedbacks* ou ajuda. A vantagem que a ação de editar oferece ao contexto educacional é que questões como idioma, regionalismo ou mesmo *feedbacks* mal elaborados não servirão de obstáculos para que o recurso seja utilizado em outra situação ou contexto.

Já a ação de **substituir** visa possibilitar a permuta de um elemento por outro de mesmo valor. Os elementos que podem sofrer esse tipo de intervenção, são: imagem, áudio e vídeo.

Por fim, a ação de **fragmentar** está vinculada somente ao elemento vídeo, pois esta ação pode ser utilizada sempre que for necessário a utilização de apenas um trecho de vídeo para trabalhar conceito. Vale lembrar que todas as ações de customização descritas nesta seção estão relacionadas ao grau de liberdade, como apresentado no Quadro 4.2.

Para uma melhor compreensão de como as ações de customização atuam em um OAC, é apresentado nas Figuras 4.1 e 4.2 o OAC Histórias Fantásticas e sua forma de adaptação com base no grau de liberdade do indivíduo.

Quadro 4.2: Ações de Customização x Grau de Liberdade

<b>Ações de Customização</b>	<b>Grau de Liberdade</b>
Desabilitar/Habilitar Esconder/Mostrar	1
Desabilitar/Habilitar Esconder/Mostrar Fragmentar	2
Desabilitar/Habilitar Esconder/Mostrar Fragmentar Editar Substituir	3
Desabilitar/Habilitar Esconder/Mostrar Fragmentar Editar Substituir Acesso ao Código Fonte do OAC	4

Fonte: Própria Autora

Considerando o Quadro 4.1, o Histórias Fantásticas é composto por elementos do tipo texto, imagem e botão. Desse modo, os tipos de intervenções que esses elementos podem sofrer são: desabilitar, esconder e editar. Sendo esta última ação, específica para o elemento texto.

Ciente do que pode ser alterado e de como essa alteração pode ser realizada, o cliente terá a liberdade, conforme GL no Capítulo 3, para efetivar as mudanças necessárias com base em sua necessidade e conhecimento. Sendo assim, para que as alterações realizadas pelo docente sejam direcionadas, os OAC dispõem de *login* e senha, conforme apresentado na Figura 4.1, que restringem a permissão para alterar o comportamento dos elementos.

Dessa forma, caso o docente tenha apenas o GL 1, ele poderá somente desabilitar ou esconder algum elemento da interface do recurso. Assim, ao se logar, o recurso é disponibilizado ao cliente, como apresentado na Figura 4.2, com os botões e imagens passíveis de serem desabilitados. A disponibilização dessas ações pode ser visualizada na Figura 4.2 através dos botões de customização presentes em cada componente da interface. Os botões habilitados indicam as operações disponíveis em cada um dos elementos. No exemplo em questão estão circulos na Figura 4.2 os botões que permitem

Figura 4.1: Tela de Login do OAC Histórias Fantásticas



Fonte: Própria Autora

desabilitar as imagens.

Figura 4.2: OAC Histórias Fantásticas com Grau de Liberdade 1



Fonte: Própria Autora

Conforme apresentado no Quadro 4.2, além da liberdade de desabilitar, o cliente com GL 1 tem ainda a possibilidade de esconder um elemento. Entretanto, como mostra a Figura 25, os botões de acesso a essa ação não estão habilitados. Isso ocorre pelo fato dessas ações não terem sido disponibilizadas pela equipe de desenvolvimento do OAC. Portanto, mesmo tendo GL suficiente para executar essa ação, não houve liberação da mesma neste recurso.

Se o docente possuir GL 3 ele poderá optar também por editar um texto. A Figura 4.3 apresenta o OAC disponibilizado a um docente com GL 3. Nela pode ser observado que também está habilitado o botão de edição na barra de customização do elemento texto

Da mesma forma, um docente com GL 3 também possui liberdade para esconder ou substituir um elemento. Mas isso, assim como a ação de esconder do GL 1, só será possível, se essa ação tiver sido fornecida ainda na fase de projeto do OAC.

Figura 4.3: OAC Histórias Fantásticas com Grau de Liberdade 3



Fonte: Própria Autora

As ações de editar um texto e/ou desabilitar uma atividade possibilitam a abertura do conteúdo do OAC Histórias Fantásticas para que os professores desenvolvam sua autonomia na criação de novas estratégias pedagógicas com esse recurso. Como mencionado anteriormente, as ações de editar um texto e/ou desabilitar/habilitar uma atividade possibilita a abertura do conteúdo e do recurso para que os professores desenvolvam sua autonomia na criação de novas estratégias pedagógicas.

É importante salientar que o impacto proporcionado pelo OAC no grau de autonomia do professor só será efetivo se as permissões para executar as ações de customização forem atribuídas a cada um dos elementos da interface do recurso, ainda na fase de desenvolvimento do mesmo.

Assim, como apresentado em Souza et al. (2007), mas do que valorizar a construção da autonomia é necessário ter ferramental que efetive essa autonomia e, consequente-

mente, contribua com a produção de ambientes que fascinem e motivem os alunos.

Sendo assim, é apresentado na subseção 4.1.4 as vantagens, juntamente com as limitações, em se customizar um OA.

#### 4.1.4 Vantagens e Limitações

Durante vários anos tem havido um interesse mundial relacionado ao compartilhamento de recursos educacionais em conformidade com a ideia de que o conhecimento deve ser disponibilizado livremente aos professores (NGUYEN et al., 2011). Nesse sentido, uma das vantagens dos OAC está relacionada ao fato deles emergirem essa ideia de compartilhamento ressaltando a importância dos professores de obterem não somente o recurso, mas principalmente a capacidade de realizarem novas configurações no cenário que lhe é apresentado, como forma de aproximar, cada vez mais, o aluno do conceito abordado.

Essa necessidade de compartilhar de modo livre o conhecimento é decorrente da mudança no comportamento educacional dos professores e alunos que tornam evidente a disfuncionalidade e ineficácia da abordagem universal (*one-size-fits-all*), no âmbito educacional (KORRES; GARCÍA-BARRIOCANAL, 2008).

Desta forma, a mudança proposta pelos OAC através de adaptações, abre a possibilidade para que novas configurações dos recursos possam ser geradas pelos próprios professores. Assim, as necessidades dos professores são colocadas em primeiro plano com os OAC. Essa abordagem é um diferencial, visto que, segundo (PETRONI; SOUZA, 2009), esse é um profissional que não é escutado e frequentemente colocado em xeque.

Diante da compreensão de um OAC para o contexto educacional e suas implicações na prática docente, outras vantagens podem ser listadas, como por exemplo a ampliação da capacidade de reutilização do recurso, maior alcance de abertura além de um maior sincronismo entre o planejamento e o OA utilizado.

No que concerne às limitações, é possível destacar a preocupação em minimizar a interferência do cliente no recurso, como forma de resguardar o conceito abordado no

mesmo. No entanto, não há garantia que possíveis problemas educacionais ou mesmo de navegação, possam ser evitados com essa interferência.

Além disso, outro aspecto relevante está relacionado à própria utilização do OAC, pois as limitações referentes à utilização do OA pelos professores podem ser herdadas pelo OAC, acrescido agora do aspecto de customização. Desta forma, para que um professor consiga customizar um OAC é necessário antes que ele compreenda o funcionamento do recurso e a partir daí será possível executar o seu planejamento no mesmo, através das ações de customização.

Diante das limitações, ainda assim a utilização dos OAC trazem ganhos substanciais ao contexto educacional. Dessa forma, é importante ter ciência não apenas em como utilizá-lo, mas também em como ocorre a sua construção. Sendo assim, é apresentado na seção 4.2 o processo de desenvolvimento bem como as estratégias adotadas na produção desse tipo de recurso.

## 4.2 Como Construir Objetos de Aprendizagem Customizáveis

O desenvolvimento de um OAC possui as mesmas necessidades que a produção de um OA monolítico, ou seja, a participação de uma equipe multidisciplinar juntamente com a utilização das documentações geradas pelos membros dessa equipe.

No entanto, é necessário observar que tanto o processo de desenvolvimento quanto a metodologia adotada na produção de OAC, requerem mudanças para tornar mais efetivos os ganhos proporcionados pelo recurso adaptável. Isto porque se faz necessária uma maior integração entre os profissionais envolvidos no que diz respeito ao ferramental empregado no desenvolvimento desse novo tipo de recurso.

Esse ferramental deve primar pela qualidade na comunicação entre os profissionais e com isso os recursos produzidos a partir dessa abordagem isentam *designers* e programadores a realizar, cada um, sua própria interpretação do que foi especificado pelos pedagogos e especialistas de área.

Uma alternativa para amenizar as limitações referentes à comunicação e a qualidade do recurso produzido diz respeito à mudança na perspectiva de desenvolvimento adotada. Para isso, era necessário que todos os membros da equipe multidisciplinar despendessem mais tempo discutindo sobre o domínio da aplicação para só depois focar na implementação do recurso. Essa mudança de foco resulta em uma maior aproximação da equipe e uma consequente melhoria na comunicação. No entanto, para que essa melhoria na comunicação fosse eficaz, era importante que todas as decisões fossem registradas em um documento que pudesse ser compreendido e compartilhado por todos. Nesse sentido, foi adotada nesta tese o desenvolvimento de OAC utilizando a estratégia de Engenharia de Software Orientada a Modelos (MDSE).

A adoção dessa estratégia pode ser justificada pelo fato desta possuir ferramental necessário para suprimir as deficiências descritas e, além disso, a MDSE faz uso de Linguagem Específica de Domínio (DSL) que por sua vez ajuda a contemplar as diferentes visões de cada um dos profissionais envolvidos.

Dessa forma, antes de apresentar de que modo o processo de desenvolvimento de um OAC é organizado, é necessário apresentar a metodologia que deu suporte a esse processo. Sendo assim, será apresentado na seção 4.2.1 a engenharia de software orientada a modelos (MDSE) e os ganhos que essa metodologia oferece para o desenvolvimento de um OAC.

### 4.2.1 Engenharia de Software Orientada a Modelos

A Engenharia de Software Orientada a Modelos (*Model-Driven Software Engineering* (MDSE)) (SCHMIDT, 2006) vem obtendo uma grande atenção pelos pesquisadores da área de Engenharia de Software nos últimos anos. Também sendo denominada de *Model-Driven Engineering* (MDE), *Model-Driven Development* (MDD), dentre outras, essa abordagem se baseia no uso sistemático de modelos como artefatos principais no processo de desenvolvimento de software. Assim, há uma substituição no foco, do desenvolvimento para os modelos, que são artefatos mais abstratos do que os códigos. Com essa abordagem, os engenheiros de software e desenvolvedores são levados a despende esforços no espaço do problema em vez de plataformas específicas. Dessa forma,



o objetivo principal da MDSE é a separação entre a lógica da aplicação e a tecnologia alvo de desenvolvimento, através da criação de modelos.

Os modelos são representações abstratas de um determinado conhecimento. No entanto, com a MDSE, os modelos podem ser descritos através de Linguagens Específicas de Domínio (DSL - *Domain-Specific Languages*). Essas linguagens possuem representações direcionadas para um público específico (KELLY; TOLVANEN, 2008). A partir dessas especificações, a MDSE sugere a aplicação de um conjunto de transformações nesses modelos, que podem ser direcionados para uma arquitetura específica, nas quais códigos podem ser gerados.

A MDSE tem o potencial de melhorar as práticas correntes de desenvolvimento de software. Segundo Yusuf, Chessel e Gardner (2006), os seguintes benefícios podem ser alcançados com a utilização desta abordagem:

- **Melhoria na Produtividade:** redução de custos de desenvolvimento de software, através da geração de código e de artefatos a partir de modelos, aumentando também a produtividade de quem desenvolve.
- **Adaptabilidade:** a adição ou modificação de uma função de negócio é linear. Quando se pretende adicionar novas funções de negócio, é necessário desenvolver apenas o comportamento específico dessa função, sendo que as tarefas restantes necessárias para se gerar os artefatos de implementação são capturados pelas transformações.
- **Consistência:** a aplicação manual de práticas de codificação e de decisões em nível de arquitetura é propícia a erros. A MDSE assegura que os artefatos são gerados de forma consistente.
- **Repetibilidade:** a MDSE é especialmente poderosa quando aplicada em nível de organização. O retorno do investimento a partir do desenvolvimento de transformações, aumenta à medida que estas são reutilizadas. A utilização de transformação experimentadas e testadas também aumenta a previsibilidade do desenvolvimento de novas funções, além de reduzir o risco, uma vez que os problemas de arquitetura e técnicos já foram resolvidos.

- Melhor comunicação com os *stakeholders*: os modelos omitem detalhes de implementação que não são relevantes para a compreensão do comportamento lógico de um sistema. Eles estão muito mais próximos do domínio do problema, reduzindo assim o *gap* semântico entre os conceitos que são compreendidos pelos *stakeholders* e a linguagem na qual a solução é expressa. Sendo assim, MDSE facilita a disponibilização de soluções que estão melhor alinhadas com os objetivos de negócio.
- Melhor comunicação do projeto: os modelos ajudam a compreender os sistemas em nível de projeto, conduzindo a uma melhoria da discussão e da comunicação sobre o sistema. Uma vez que os modelos fazem parte da definição do sistema, e não da documentação, nunca ficam desatualizados e são confiáveis.
- Captura do conhecimento: os projetos ou as organizações dependem frequentemente de especialistas que tomam as melhores decisões em termos de prática. O seu *know-how* é capturado em padrões e transformações. Assim, se as transformações forem acompanhadas de documentação suficiente, o conhecimento de uma organização é mantido nos padrões e transformações, mesmo quando os especialistas vierem a sair.
- Capacidade de prorrogar as decisões tecnológicas: o desenvolvimento inicial de aplicações é focalizado nas atividades de modelagem e especificação. É possível prorrogar as escolhas de uma plataforma ou produto tecnológico específico até se ter disponível mais informações. Em áreas com ciclos de desenvolvimento extremamente longos (como os sistemas de controle de tráfego aéreo), este aspecto é crucial.

Dentre as abordagens atuais para o tratamento de MDSE se destacam a *Model-Driven Architecture* (MDA) (SOLEY et al., 2000) e o *Eclipse Modeling Framework* (EMF) (MERKS; GROSE, 2003). A MDA é uma especificação para apoiar a MDSE que possui uma arquitetura própria e um conjunto de estratégias definidas pela *Object Management Group* (OMG) fortemente baseada na *Unified Modeling Language* (UML). A arquitetura MDA utiliza o conceito de três níveis de modelagens, indo do mais genérico ao mais específico. Os níveis são os seguintes:

- CIM (*Computational Independent Model* ou Modelo Independente de Computação), que representa o nível mais alto de abstração, voltado para a representação dos requisitos.
- PIM (*Platform Independent Model* ou Modelo Independente de Plataforma), que apresenta um grau mais de especificação, representando o sistema do software, mas de uma maneira independente de qualquer plataforma.
- PSM (*Platform Specific Models* ou Modelo Específico de Plataforma) é o nível de modelo onde são acrescentadas as especificações da plataforma para a qual o modelo será transformado. Deve existir um PSM diferente para cada plataforma.

Além desses conceitos, a MDA também especifica formatos e linguagens padrões:

- MOF (*Meta-Object Facility*): É o meta-metamodelo padrão do MDA. Ele define uma interface padrão de acesso aos dados dos modelos e regras para a criação de interfaces específicas.
- QVT (*Queries/Views/Transformations*): É a linguagem de transformação de modelos padrão do MDA. Ela define consultas e transformações de modelos baseadas no MOF.
- XMI (*XML Metadata Interchange*): É o formato, baseado em XML, para representar de forma textual os modelos compatíveis com o MOF.

Já o EMF é baseado no meta-metamodelo Ecore, em contraposição ao MOF. Ecore é utilizado por todos os projetos desta abordagem. A abordagem Eclipse abriga um grande número de projetos e ferramentas de desenvolvimento voltadas para a MDSE. Algumas destas são :

- GMT (*Generative Modeling Tools*): É uma incubadora para projetos de ferramentas, linguagens e frameworks voltados para os conceitos da MDSE. Dentre os projetos atuais da GMT, pode-se citar os seguintes:
  - ATL (*Atlas Transformation Language*): É uma linguagem de transformação de modelos equivalente ao QVT, mas que tem como base o Ecore.

- MOFScript: É uma linguagem de transformação de modelos para texto (M2T).
- TCS (*Textual Concrete Syntax*): Permite o desenvolvimento de linguagens de domínio específico (DSLs) textuais.
- GMF (*Graphical Modeling Framework*): Ferramenta voltada para a criação e manipulação de linguagens visuais de domínio específico.
- JET (*Java Emitter Templates*): Ferramenta de geração de códigos baseada em *templates*.

Para esta tese, a abordagem do ECLIPSE foi adotada para o desenvolvimento de OAC. Essa escolha se deu, além da vasta coleção de *plugins* para a criação de DSL visuais, pela necessidade de utilizar ferramentas de livre acesso, haja vista que a filosofia dos OAC está relacionada ao compartilhamento de ideias e conhecimento.

Como forma de mediar o desenvolvimento de OAC a partir da perspectiva do MDSE, é apresentado na subseção 4.2.2 um processo de desenvolvimento proposto nesta tese, que dá suporte à criação dos OAC, denominado de CLAssRoOM.

## 4.2.2 Processo CLAssRoOM

O *Customizable LeARning Objects Model-driven* (CLAssRoOM)<sup>1</sup> é um processo de desenvolvimento que faz uso da abordagem orientada a modelos associado à estratégia da customização guiada para produção de OAC.

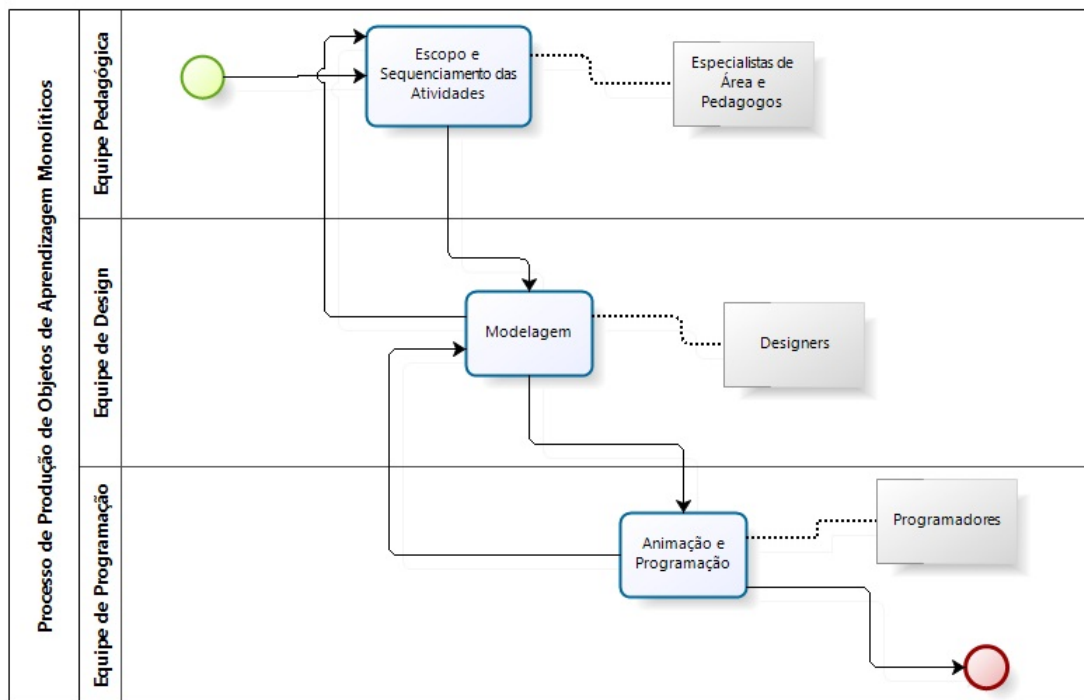
De acordo com a Figura 4.4, o processo de produção de objetos não customizáveis (OA Monolíticos) é executado de modo similar ao processo de um OAC apresentado na Figura 4.5. Dessa forma, independente do tipo de OA desenvolvido, ambos os processos utilizam a mesma estrutura de equipe.

Em um processo de produção de OA monolíticos os especialistas de área são responsáveis pela definição do escopo e, juntamente com os pedagogos, também definem a estrutura do sequenciamento das atividades. Essa estrutura é organizada em um documento que recebe o nome de Roteiro de Atividades e serve de base para os *designers*

---

<sup>1</sup>Esse foi um acrônimo gerado pela ferramenta *Acronym Creator* (<http://acronymcreator.net/>)

Figura 4.4: Processo Genérico de Produção de um OA Monolítico



Fonte: Própria Autora

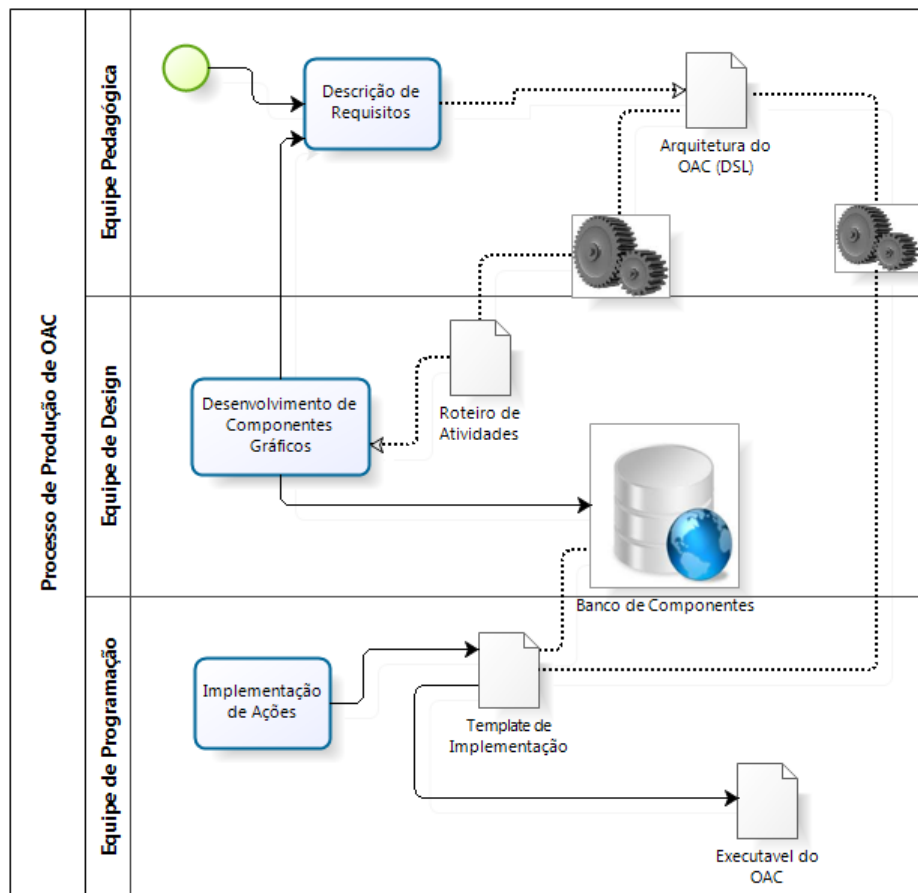
criarem o projeto gráfico. Este mesmo documento também servirá de suporte para os programadores fazerem a animação e a programação dos OA.

O CLAssRoOM, por sua vez, difere por utilizar a abordagem da MDSE para a criação e manipulação de modelos como artefatos principais no processo de desenvolvimento de software (SOUZA; CASTRO-FILHO; ANDRADE, 2011b). Artefatos estes que são desenvolvidos com base em uma *Domain Specific Language* (DSL) que fornece subsídios para a construção da arquitetura dos OAC.

Para uma melhor compreensão da dinâmica do processo de produção de um OAC, bem como dos artefatos gerados nele, é apresentado na Figura 4.5 o fluxo de funcionamento do CLAssRoOM.

Por adotar uma estratégia orientada a modelos, no CLAssRoOM há uma concentração de esforços na etapa de concepção, de modo que o esforço na fase de implementação seja reduzido, sendo que a preocupação é direcionada em “o que desenvolver” e não em “como desenvolver” (SOUZA; CASTRO-FILHO; ANDRADE, 2010b; SILVA; BARBOSA; MALDONADO, 2011).

Figura 4.5: Processo de Produção de um OAC



Fonte: Própria Autora

Este processo possibilita a especificação, em termos de elementos de arquitetura, dos modelos das interfaces que irão compor o OAC e as relações de dependência existentes entre elas, o que constitui a arquitetura do recurso. Para tanto, é utilizada uma DSL que possibilita a criação desses modelos que serão utilizados, posteriormente, nas etapas de desenvolvimento e implantação dos OAC.

A produção de um OAC é executada por uma equipe multidisciplinar composta por pedagogos, especialistas de área, *designers* e programadores, sendo cada um desses profissionais responsáveis pela construção dos artefatos gerados durante o processo.

Como apresentado na Figura 4.5, o CLassRoOM compreende 3 etapas. São elas: Descrição dos Requisitos, Desenvolvimento de Componentes Gráficos ou simplesmente Modelagem e, por fim, a Implementação de Ações. Em cada uma dessas etapas é gerado um ou mais artefatos que serão utilizados durante o desenvolvimento de um

OAC.

O processo inicia com a definição do escopo pelo especialista de área e a descrição dos requisitos pela equipe pedagógica. Com base no que foi descrito a arquitetura do OAC é especificada pelos membros da equipe. Essa especificação é construída através da notação fornecida pela DSL.

Essa arquitetura produzida pela DSL, serve de base para gerar, via regras de transformações, tanto o Roteiro de Atividades, a ser utilizado pela equipe de *design*, quanto os *templates* de implementação, a serem utilizados posteriormente pela equipe de programação. O roteiro de atividades de um OAC difere do OA monolítico devido a forma como são produzidos, pois enquanto no OAC ele é gerado pelo CLAssRoOM a partir do modelo de arquitetura criado com o suporte da DSL, no OA monolítico ele é criado manualmente, pelos especialistas de área e pedagogo, a partir de descrições feitas com base no escopo e no sequenciamento das atividades.

No caso do OAC, o roteiro de atividades, como o próprio nome indica, serve para nortear a modelagem do conjunto de interfaces que compõem o recurso. Com base no roteiro especificado, a equipe de *design* desenvolve os componentes gráficos que irão compor o recurso. Componente gráfico deve ser entendido como todo elemento que possui algum tipo de relação com a interface do recurso. Os componentes gráficos modelados são armazenados em um banco de componentes para que a equipe de programação possa utilizá-lo posteriormente.

Outra diferença entre o processo de desenvolvimento de um OAC e de um OA monolítico é que no primeiro há uma preocupação em se modelar componentes independentes, favorecendo assim a reutilização dos mesmos. Enquanto nos OA monolíticos, não há uma clara preocupação nesse tipo de reutilização, haja vista que não se projeta os componentes de modo independente.

Com relação aos *templates* de implementação, a equipe de programação irá fazer uso desses artefatos para realizar alguns ajustes, tais como a disposição dos elementos na tela do recurso, ou adaptações no código. Esses documentos são arquivos que possuem os códigos em uma linguagem de programação alvo, que contém as informações dos componentes descritos na DSL e as ações relacionadas a esses componentes, tais como:

ações de customização, ações de navegação, evento de mouse, dentre outras.

Uma vez finalizado os ajustes, as imagens que estão no banco de componentes são invocadas, para juntos, imagens e ações, darem origem ao executável do OAC.

Levando em consideração a descrição do processo de produção de um OAC, será apresentada em detalhes nas próximas subseções, cada um dos artefatos gerados durante esse processo.

#### 4.2.2.1 Modelo de Arquitetura

A arquitetura de um software, como descrito em Bass, Clements e Kazman (2003), pode ser compreendida como sendo um conjunto de elementos que compõe sua arquitetura e que possui alguma organização, sendo que essa organização rege as relações entre esses elementos. Já em Perry e Wolf (1992), arquitetura de *software* é definida como sendo a estrutura ou estruturas de um sistema, a qual é composta por elementos de software, propriedades externamente visíveis desses elementos e relacionamentos entre eles

Como pode ser observado, a definição da estrutura da arquitetura de um software se preocupa com a exploração da organização de alto nível desse software, possibilitando a realização de inferências sobre o comportamento que o mesmo terá depois de sua implementação. Contudo, como consequência da existência de diversos interessados nos objetivos a serem alcançados pelo software, a arquitetura deve possuir diversas visões, que devem explorar o software através de perspectivas específicas da arquitetura. No caso dos OAC, a visão de navegação deve ser explicitada, visto que as transições entre elementos que os compõem influenciam diretamente na qualidade pedagógica deste tipo de recurso.

Sendo assim, no desenvolvimento dos OAC, a descrição da arquitetura deve permitir, além de explicitar a organização desse tipo de software, facilitar a transição da etapa de requisitos para o projeto detalhado e, posteriormente, a sua implementação.

Dessa forma, para uma melhor compreensão, serão considerados os componentes básicos estruturais tanto de um OA quanto de um OAC como sendo as Cenas. Uma Cena deve ser entendida como uma unidade de arquitetura que, por sua vez, pode ou



não estar relacionada com outras Cenas através de um conjunto de ações. Na Figura 4.6, por exemplo, são apresentadas Cenas do OA Pontos em Batalha<sup>2</sup>. Nela, ao ser selecionada a opção “Início”, na Cena que fica do lado esquerdo da figura, deve haver uma transição para a Cena apresentada no lado direito.

Figura 4.6: Cenas do OA Pontos em Batalha



Fonte: <http://proativa.virtual.ufc.br/oa/pontos/pontos.html>

De forma a organizar a estrutura topológica do OAC, cada Cena pode possuir uma estrutura hierárquica interna composta por outras Cenas, denominadas nesse caso de subcenas. Essa composição possibilita identificar Cenas que possuem transições internas nas quais as interfaces sofrem pequenas alterações, como a apresentação de *pop-ups*, ou a troca de posição de componentes de interface, sem, no entanto, estar relacionada à transição para outra Cena. Na Figura 4.7, pode ser visualizada uma Cena interna ou subcena do OA Pontos em Batalha na qual são apresentadas as estatísticas de tentativas do aluno.

Figura 4.7: Cena Interna do OA Pontos em Batalha

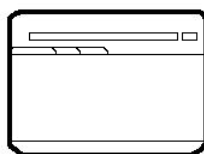


Fonte: <http://proativa.virtual.ufc.br/oa/pontos/pontos.html>

<sup>2</sup><http://proativa.virtual.ufc.br/oa/pontos/pontos.html>

Para descrever o modelo de arquitetura dos OAC, deve ser utilizada uma representação diagramática, proposta inicialmente em Souza, Castro-Filho e Andrade (2010a), que identifica as Cenas e as Transições. As Cenas devem ser representadas pela figura de um Browser, como indicado na Figura 4.8.

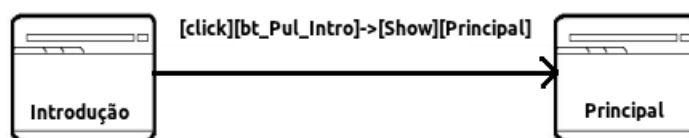
Figura 4.8: Representação da Arquitetura de uma Cena



Fonte: Própria Autora

As transições devem ser indicadas por setas direcionais. Essas setas podem, eventualmente, ter uma identificação na forma de legendas, que apresentam informações sobre os elementos que originaram as ações e sobre o destino dessas ações. A Figura 4.9, apresenta a representação da transição da Cena “Introdução” através do evento “click” realizado sobre o componente de interface identificado por “*bt\_Pul\_Intr*”. Esse evento, por sua vez, leva a ação “*Show*” da Cena “Principal”.

Figura 4.9: Representação de uma Transição



Fonte: Própria Autora

#### 4.2.2.2 Arquitetura de Interface de Cenas

As transições podem necessitar de informações dos elementos que formam as interfaces das Cenas (Figura 4.9). Além disso, a construção da interface de uma Cena é uma tarefa que deve ser realizada de modo preciso, visto que as decisões sobre as representações das informações nessas Cenas influenciam tanto na qualidade visual quanto pedagógica de um OA. Um bom projeto de interface para cada Cena influencia diretamente na qualidade das etapas subsequentes, visto que as decisões pedagógicas e suas implicações podem ser realizadas de forma abstrata logo no início do processo de desenvolvimento.

Desta forma, é introduzido no modelo de arquitetura dos OAC, o detalhamento de cada Cena. Esse detalhamento é realizado através da descrição da arquitetura das interfaces dessas Cenas. Observando que cada Cena é composta por um conjunto de elementos simples, é possível descrevê-las também de modo abstrato. Assim, da mesma forma que *Wireframes*<sup>3</sup> (SNYDER, 2003) são utilizados na descrição de páginas *Web*, foi aplicado esse conceito para a descrição do OAC, na qual são identificados um conjunto de elementos de interface que devem ser utilizados nesta tarefa.

#### 4.2.2.3 Roteiro de Atividades

O Roteiro de Atividades (ou simplesmente Roteiro ou *Storyboard*) é um documento desenvolvido para auxiliar *designers* e programadores a compreender melhor toda a estrutura do OAC que foi projetada pela equipe pedagógica, uma vez que o mesmo possui, além das representações do modelo de arquitetura, descrições detalhadas sobre os cenários, ações e *feedbacks* presentes em cada uma das cenas.

Como mencionado anteriormente, é importante ressaltar que este documento não é uma particularidade do processo de desenvolvimento proposto nesta tese, haja vista sua existência em qualquer processo de desenvolvimento de OA. No entanto, o roteiro de atividades do OAC difere do roteiro do OA monolítico dentre outras coisas, devido a existência de uma representação gráfica, juntamente com a descrição do modelo de arquitetura presentes no documento.

As descrições detalhadas sobre cada uma das cenas presentes nos roteiros tem por objetivo explicitar comportamentos que não estão presentes no modelo de arquitetura pelo nível de detalhamento que os mesmos apresentam, como, por exemplo, animações, descrição sobre o tipo de imagem a ser utilizado levando em consideração o nível de escolaridade em que o recurso será aplicado, informações sobre *feedbacks* positivos e negativos, ajuda, fala de personagens, além dos tipos de dados a serem inseridos nos recursos.

Considerada uma ferramenta importante em termos de documentação no processo

---

<sup>3</sup>Desenho básico, como um esqueleto, que demonstra de forma direta a arquitetura de como o objeto (interface, página da internet, modelo, etc.) final será de acordo com as especificações relatadas.

de desenvolvimento, o roteiro serve, principalmente, para facilitar a interação entre todos os envolvidos no processo de produção do OAC ou seja, equipe pedagógica, *design* e programação.

Para exemplificar precisamente o que de fato este documento especifica, a Figura 4.10 apresenta alguns tópicos relacionados ao Roteiro do OA Pontos em Batalha, gerados pela equipe pedagógica, e que foram projetados e implementados pelas equipes de *design* e programação.

Figura 4.10: Fragmento de um Roteiro de Atividades

1. As coordenadas da caravela são fornecidas, cabendo ao usuário descobrir as coordenadas da embarcação pirata;
2. Descoberta as coordenadas da embarcação pirata, o usuário deverá informá-la na caixa “localização dos barcos no mar”;
3. Inserido o valor de X e de Y da embarcação pirata na caixa “localização dos barcos no mar” o usuário deverá efetuar o cálculo da distância entre os dois pontos;
4. Caso o usuário não saiba como efetuar esse cálculo, o objeto disponibiliza um ícone “ajuda” que fica no campo superior a direita da tela e é simbolizado por um farol;
5. Calculada a distância, o usuário deverá inserir o resultado do seu cálculo na caixa “distância da trajetória da bala”;
6. Imediatamente, após inserir o valor na caixa especificada no passo 5, será fornecido o botão “atirar” em que o usuário irá verificar se seu cálculo está correto
7. O botão “estatística” pode ser utilizado para mostrar uma tela com estatísticas de erros e acertos nos tiros

Fonte: Própria Autora

É importante ressaltar que a escolha pelo roteiro do recurso Pontos em Batalha se deu pelo fato do mesmo ter sido utilizado para exemplificar o modelo de arquitetura.

Devido ao nível de detalhamento que o documento Roteiro de atividades apresenta, é possível afirmar que este artefato gerado com base no modelo de arquitetura é de fundamental importância para os profissionais envolvidos no processo de produção do OAC.

#### 4.2.2.4 *Templates* de Implementação

Dentro do processo de produção do OAC é gerado o artefato de implementação, além dos relacionados ao modelo de arquitetura e do roteiro de atividades, denominado de *templates* de implementação.

Estes documentos visam auxiliar os programadores no desenvolvimento do código do recurso, pois eles têm por finalidade descrever, utilizando a linguagem de programação adotada, os códigos de implementação de cada um dos componentes projetados pelo modelo de arquitetura.

De uso restrito da equipe de programação, os *templates* de programação foram gerados, dentro da abordagem orientada a modelos, para que os programadores agilizem a produção do executável do OAC. Deste modo, grande parte do código é gerado dinamicamente, cabendo à equipe de desenvolvimento realizar adaptações necessárias para detalhar algumas das ações descritas em nível de arquitetura.

Para que a geração dos artefatos descritos seja compreendida, serão apresentadas as regras de transformação necessárias para a geração dos mesmos.

#### 4.2.2.5 Regras de Transformações

As regras de transformações são normas estabelecidas para transformar a arquitetura do OAC criada em documentos passíveis de serem compreendidos tanto pelo designer (roteiro de atividades) quanto pelo programador (*templates* de implementação).

Essas regras de transformação estão associadas às informações contidas no metamodelo que regem a criação dos OAC. Essas regras realizam associações entre os elementos do OAC e as suas implementações em uma linguagem alvo.

Na código a seguir, é mostrado um exemplo geral de uma regra que escreve no documento de saída (no caso um *template* de implementação) uma função em linguagem de programação. Deve ser observado que existem elementos que serão escritos diretamente na saída, como o caso das palavras “*Function*” e “:void{” e outros que serão avaliados junto à arquitetura, no caso as variáveis “cena” e “componente”, sendo que os respectivos valores serão substituídos no momento da execução da transformação.

```
escrever("Function" + cena + "." + componente + ":void{");
```

Após ser executado, o *template* pode gerar o seguinte trecho de código a partir da substituição desses valores:

```
Function principal.bt_at1:void{
```

### 4.3 Discussão

Há muito se discute sobre a necessidade de solucionar problemáticas relacionadas à produção de RED, mais especificamente de OA. Problemas esses que estão direcionados, dentre outros, à abertura do recurso, tempo de desenvolvimento, interação entre os membros da equipe e qualidade do recurso desenvolvido. Isto ocorre, porque a forma como os OA são desenvolvidos atualmente não buscam soluções efetivas para sanar ou mesmo minimizar tais problemáticas.

Sendo assim, este Capítulo apresenta um novo tipo de recurso educacional, denominado de Objeto de Aprendizagem Customizável (OAC) como uma alternativa para solucionar as problemáticas anteriormente descritas.

Os OAC surgem com a proposta de tratar essas limitações impostas pelos métodos de desenvolvimento empregados para a produção de OA, pois seu processo de desenvolvimento faz uso da engenharia de software orientada a modelo e esta, por sua vez, favorece o tratamento em nível abstrato do recurso. Esse fato, associado à estratégia da customização guiada, possibilita que diversas alterações nos elementos que compõem a interface do recurso, como por exemplo, editar, substituir, esconder, desabilitar e fragmentar, possam ser realizadas na interface do OAC por qualquer usuário.

Para facilitar o desenvolvimento de OAC é apresentado neste capítulo o processo de produção denominado CLAssRoOM, que serve para ilustrar de que modo esse tipo de recurso deve ser produzido, bem como a documentação e os artefatos gerados a partir de sua utilização.

Para uma melhor compreensão da relação entre as estratégias de desenvolvimento de OA tradicionais e a proposta nesta tese, é apresentado no Quadro 4.3 de que modo os OAC e os OA enxergam e tratam cada problemática levantada neste Capítulo.

Os ganhos adquiridos com o desenvolvimento do OAC, e sua consequente utilização, está no fato de que tanto desenvolvedores quanto professores, terão liberdade para atuar de modo efetivo nas esferas de suas competências, ou seja, os desenvolvedores terão acesso aos componentes do OAC, juntamente com os fontes para realizar revisões ou mesmo a aplicação do próprio reuso para o desenvolvimento de outro OAC. Sendo assim, os professores não precisarão mais supor uma determinada situação ou contexto, pois com o OAC ele poderá fazer efetivamente ajustes ou adaptações no mesmo que viabilizem, além do reuso, uma reproposta (*repurpose*) (GUNN; WOODGATE; O'GRADY, 2005) de um contexto ou situação planejado, fazendo com que o professor assuma o controle de suas práticas e aproveite, de fato, o potencial transformador da tecnologia para a educação.

## 4.4 Conclusão

Devido ao volume de OA desenvolvidos nos últimos anos, a preocupação agora deve estar centrada na busca por meios que ampliem a capacidade de reutilização desses recursos, tanto por desenvolvedores quanto por professores, sem no entanto, limitar a competência de nenhum desses profissionais

Um dos meios encontrados mais difundidos está relacionado à disponibilização dos códigos fonte. Essa alternativa, apesar de favorecer a reutilização do OA por parte dos desenvolvedores, oferece pouca condição aos professores de executar qualquer tipo de ajuste, haja vista que para fazer alterações nos códigos fonte é necessário que o indivíduo tenha conhecimentos específicos, que na maioria das vezes, fogem da área de domínio dos docentes.

Sendo assim, este capítulo apresentou o CLAssRoOM como um processo de desenvolvimento que faz uso da abordagem orientada a modelos e da estratégia de customização guiada aplicada à produção de OA, no intuito de amenizar as limitações

Quadro 4.3: Comparação OA x OAC

<b>Característica</b>	<b>OA</b>	<b>OAC</b>
<b>Comunicação entre membros da equipe</b>	A não sistematização das ações contribui com o surgimento de muitos “ruídos” de comunicação	Adoção de documentação como elemento principal do processo de desenvolvimento permite a integração da equipe, o que melhora a comunicação
<b>Abertura do recurso (modificações)</b>	Toda e qualquer modificação a ser feita neste tipo de recurso, só pode ser realizada em nível de código o que resulta no alcance de uma pequena parcela de pessoas aptas a modificá-lo	Todo e qualquer indivíduo terá condições de realizar as adaptações necessárias no recurso, através da estratégia da customização guiada
<b>Reuso</b>	Há distinção entre o reuso dos fontes e o reuso em outros contextos. Mais comum o reuso em outros contextos, haja vista que os fontes não são, em geral, facilmente disponibilizados	Realizado nas esferas educacionais e técnica, pois através dos graus de liberdade qualquer indivíduo, de acordo com seu interesse e conhecimento, tem condições de realizar ajustes na interface do recurso.
<b>Autonomia Docente</b>	Docente não interfere no recurso	O docente tem liberdade de ajustar o recurso ao seu planejamento
<b>Documentação</b>	Geralmente produzida em linguagem natural	Com o auxílio do MDSE e do uso de DSL visual a documentação é gerada e possui além da representação gráfica textos explicando detalhes do que foi projetado
<b>Desenvolvimento</b>	Existem diversos processos, mas não há evidências de ferramentas a ser utilizada pela equipe como um todo para auxiliar no processo de desenvolvimento	O CLAssRoOM é implementado por uma ferramenta de autoria desenvolvida especificamente para auxiliar as equipes de desenvolvimento na produção de OAC

Fonte: Própria Autora



---

existentes, principalmente para os professores, na realização de ajustes ou adaptações no recurso. Os OA desenvolvidos a partir desse processo são denominados de Objeto de Aprendizagem Customizável (OAC)

Vale ressaltar, que a referida estratégia aplicada à produção dos OAC, visa democratizar, entre os profissionais mencionados, a capacidade de ajustes no recurso, buscando respeitar os limites e a área de conhecimento de cada um deles.

## Capítulo 5

# CLO Studio: Ferramenta de Autoria para Produção de OAC

“First, solve the problem. Then, write the code.”  
– *John Johnson*

O conceito de Customização Guiada aplicada à produção de OA, apresentado no Capítulo 4, deu subsídios para a produção de um tipo específico de RED, denominado de OAC. Neste capítulo é apresentada a implementação do CLO Studio, que é uma DSL Visual desenvolvida com a finalidade de dar suporte ao processo CLAssRoOM.

### 5.1 Desenvolvimento do CLO Studio

Conforme apresentado no Capítulo 4, a criação de um OAC segue a abordagem da MDSE. Segundo essa abordagem, um software deve ser construído através de modelos de alto nível e os códigos devem ser gerados a partir de transformações entre esses modelos. Deste modo, por adotar essa estratégia de desenvolvimento na produção de OAC, é necessário a criação de uma linguagem que possibilite a construção de modelos específicos que venham a capturar os conceitos do domínio da aplicação. Esse tipo de linguagem, denominada *Domain Specific Language* (DSL), deve fornecer uma notação que possibilite a manipulação direta dos elementos necessários a autoria de OAC. Para dar suporte a esse processo de autoria de OAC (CLAssRoOM), foi desenvolvido neste trabalho o CLO Studio<sup>1</sup>. Essa ferramenta, construída na forma de uma DSL Visual (FOWLER; PARSONS, 2010), tem por finalidade fornecer todas as ações e notações necessárias para a autoria de OAC.

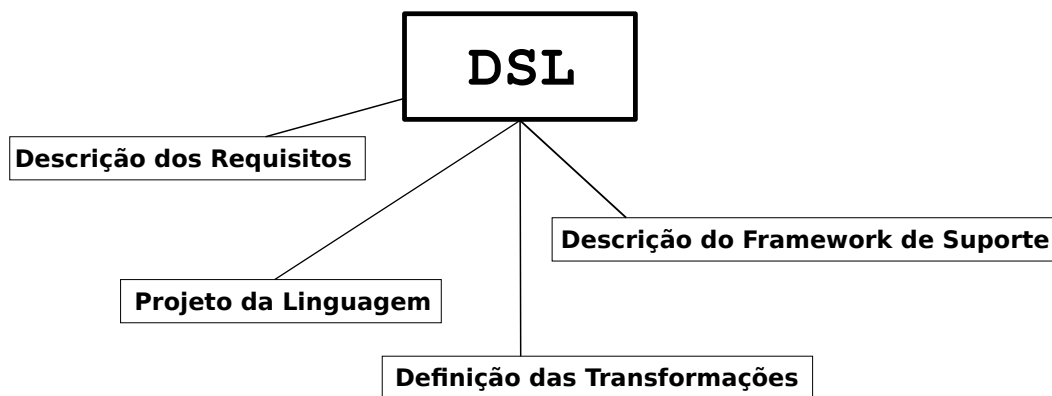
---

<sup>1</sup> O termo CLO é uma referência à *Customizable Learning Object*, que é a expressão em inglês para OAC.

Segundo Karlsch (2007) o desenvolvimento de uma DSL pode ser dividido nas seguintes fases: decisão, análise, projeto, implementação e instalação. Em Mernik, Heering e Sloane (2005) essas fases são capturadas nos seguintes padrões (“*patterns*”): padrões de decisão, padrões de análise, padrões de projeto e padrões de implementação.

De modo geral, como apresentado em Kelly e Tolvanen (2008) e adotado nesta tese, o processo de criação de uma DSL, conforme apresentado na Figura 5.1, deve responder às seguintes necessidades:

Figura 5.1: Necessidades de uma DSL.



Fonte: Própria Autora a partir de (KELLY; TOLVANEN, 2008)

1. Descrição dos Requisitos: tanto os objetivos pretendidos com a DSL como as ferramentas a serem disponibilizadas devem ser claramente identificados. Neste momento também devem ser consideradas as plataformas alvo da geração do código, visto que essa definição pode ser necessária para se compreender algumas restrições da linguagem. Além disso, também devem ser identificadas as ferramentas necessárias à implementação da DSL.
2. Projeto da Linguagem: os conceitos a serem manipulados na DSL devem ser definidos e especificados em detalhes. Para tanto deve ser descrito o metamodelo que servirá de base para a construção de modelos através da DSL. Além disso, devem ser especificadas as restrições da linguagem, através de regras de validação de modelos; a notação a ser utilizada para representar os conceitos e outras informações necessárias para descrever a sintaxe da linguagem.
3. Definição das Transformações: todas as transformações a serem utilizadas na DSL devem ser especificadas. Para isso devem ser definidas as relações entre

os modelos de origem e de destino e devem ser projetados e implementados os algoritmos que levam um modelo no outro.

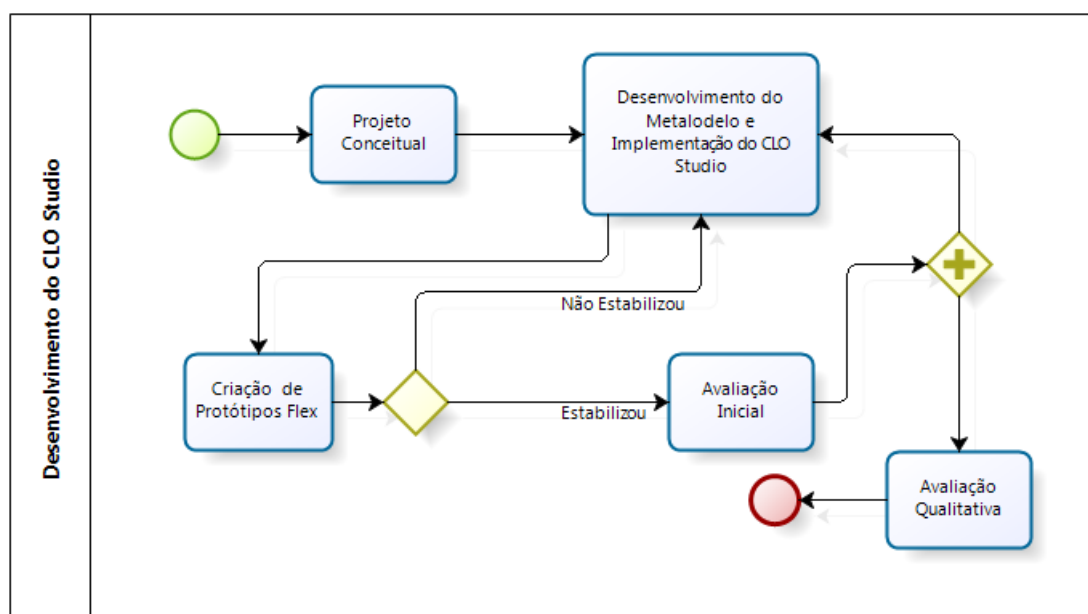
4. Descrição do *Framework* de Suporte: todos os arquivos, softwares e implementações adicionais que devem fazer parte do pacote final da DSL devem ser descritos nesta etapa. Esse *framework* não deve ser gerado a partir dos modelos, mas deve ser anexado ao produto final para que esse possa ser utilizado.

A metodologia descrita na Seção 5.1.1 mostra como essas necessidades foram tratadas no desenvolvimento do CLO Studio

### 5.1.1 Metodologia

O desenvolvimento do CLO Studio ocorreu de forma gradual. Como se espera no desenvolvimento de uma DSL, os conceitos foram sendo agregados à ferramenta de modo contínuo, e o metamodelo evoluiu para acomodar esses conceitos. Deste modo, a Figura 5.2 apresenta as etapas da metodologia utilizada na concepção e implementação dessa ferramenta.

Figura 5.2: Metodologia de Desenvolvimento do CLO Studio



Fonte: Própria Autora

- **Projeto Conceitual:** nesta primeira etapa, foi realizado um estudo sobre os elementos necessários para a composição de OAC. Esse estudo se baseou na análise de diversos OA disponíveis, principalmente, no Banco Internacional de Objetos Educacionais (BIOE) e em vários outros repositórios espalhados pela Internet. Além disso, foram analisados os documentos utilizados na criação dos OA desenvolvidos pela equipe do Proativa-UFC. Como resultado dessas análises, foi feito um primeiro esboço dos elementos visuais necessários à criação de modelos no CLO Studio. Adicionalmente, foram levantados os requisitos necessários à manipulação dos modelos pelo CLO Studio. Como resultado foram identificadas as operações de geração de código, simulação e execução de modelos.
- **Desenvolvimento do Metamodelo e implementação do CLO Studio:** de posse das informações levantadas na etapa anterior, o metamodelo do CLO Studio foi especificado e, a partir dessa especificação, foi desenvolvida a implementação dessa ferramenta. Como apresentado na Figura 5.2, essa etapa foi repetida algumas vezes, sempre considerando novas informações a serem incorporadas ao metamodelo e, por conseguinte, gerando uma nova versão do CLO Studio.
- **Criação de Protótipos de Códigos Flex e dos *scripts* de Transformação:** para que fosse avaliada a criação de aplicação com o CLO Studio, vários protótipos de OAC foram criados em Flex com a intenção de definir uma estrutura geral para essas aplicações e compreender a forma com que os conceitos do metamodelo se relacionavam com os códigos gerados. Com a evolução dessa compreensão, foi desenvolvida a primeira versão do *script* de geração de código. Como observado na Figura 5.2, essa etapa também foi repetida algumas vezes até que fosse encontrado um ponto de estabilização no qual os *scripts* de transformação estavam conseguindo gerar os códigos produzidos nos protótipos. Eventualmente o metamodelo teve que passar por ajustes de modo a facilitar a realização das transformações, tais como a definição de novos atributos e a exclusão de outros. Esses ajustes também foram incorporados à implementação do CLO Studio, fazendo com que essa ferramenta também evoluísse para acompanhar o seu metamodelo. Ainda nesta etapa, também foram definidas e implementadas as ações de customização associadas a cada tipo de elemento do metamodelo.

- **Avaliação Inicial:** depois de gerar uma primeira versão estável do CLO Studio, essa ferramenta foi avaliada de modo informal pela equipe de desenvolvimento de OA do Proativa-UFC. Também um OAC desenvolvido seguindo os moldes do CLO Studio foi submetido à avaliação de alguns professores. Essa avaliação inicial serviu para melhorar a compreensão do impacto dos resultados até aquele momento. Com o resultado dessa avaliação alguns ajustes foram realizados no CLO Studio e, em paralelo, uma avaliação mais detalhada foi especificada.
- **Avaliação Qualitativa:** a versão do CLO Studio produzida a partir da avaliação inicial foi submetida a uma avaliação qualitativa de modo a validar tanto a ferramenta como os OAC gerados. Essa avaliação é descrita em detalhes no Capítulo 6.

Como pode ser observado, o desenvolvimento do CLO Studio ocorreu de modo gradual, sendo que essa ferramenta já incorpora algumas sugestões fornecidas pelo seu público alvo. Deste modo, como se espera em uma DSL, sua evolução vem ocorrendo de modo a responder às demandas de seus clientes.

Como apresentado anteriormente, a metodologia apresentada na Figura 5.2 tem por finalidade responder às necessidades apresentadas na Figura 5.1 no contexto do CLO Studio. Assim, nas seções 5.2, 5.3, 5.4 e 5.5, são detalhadas como cada uma dessas necessidades foram tratadas no desenvolvimento dessa ferramenta.

## 5.2 Descrição dos Requisitos

### 5.2.1 Objetivos

Conforme apresentado no Capítulo 4, os aspectos de customização guiada dos OAC devem ser tratados ainda na fase de concepção desses recursos. Deste modo, os objetivos do CLO Studio estão diretamente relacionados às necessidades das equipes de desenvolvimento para tratar essa necessidade. Visto que essas equipes devem trabalhar colaborativamente para que esses aspectos de customização possam ser descritos de forma clara, os seguintes objetivos devem ser atingidos pelo CLO Studio:

- Ter uma notação simples: considerando o público alvo do CLO Studio como sendo uma equipe de desenvolvimento de OA, é importante observar os tipos de informações necessárias para que essa equipe possa compartilhar a autoria dos OAC. Desse modo, o CLO Studio deve fornecer uma notação simples e de fácil compreensão por todos os profissionais envolvidos. Por conta da multidisciplinaridade dessas equipes de desenvolvimento, é importante que seus membros possam desenvolver plenamente suas ações no processo, essa notação deve facilitar a comunicação entre esses profissionais através de abstrações que sejam compartilhadas por todos.
- Incentivar a colaboração: mesmo sendo necessário preservar as peculiaridades das atividades de cada um dos profissionais de uma equipe de desenvolvimento de OAC, o CLO Studio deve fornecer mecanismos que incentive o trabalho colaborativo sem gerar interferências entre as ações individuais.

### 5.2.2 Plataforma Alvo

A definição da plataforma alvo associada ao CLO Studio pode ser dividida em duas partes: Plataforma de Implementação do CLO Studio e Plataforma de Geração de Código. A primeira plataforma é a utilizada na implementação do próprio CLO Studio. Essa plataforma deve fornecer o ferramental necessário para que possam ser desenvolvidos todos os requisitos dessa DSL. A segunda é a plataforma de destino da geração de código do CLO Studio, que deve considerar a linguagem (ou linguagens) de programação na qual os OAC serão gerados.

A plataforma alvo para a implementação do CLO Studio foi o *Eclipse Modeling Framework* (EMF). A escolha de um subprojeto da plataforma Eclipse se deu por vários fatores. Além de ser um projeto colaborativo e de código aberto, o Eclipse é uma plataforma na qual a maioria dos desenvolvedores estão familiarizados atualmente. Além disso, o Projeto EMF disponibiliza um conjunto expressivo e maduro de ferramentas de suporte ao desenvolvimento orientado à modelos, como é o caso do padrão de descrição de metamodelos ECore(BUDINSKY, 2004), sendo que a grande maioria dessas ferramentas também são de código aberto.

Em termos de plataforma alvo para a geração de código, optou-se pela plataforma Adobe Flex <sup>2</sup>. Essa plataforma, desenvolvida pela Adobe, possui código aberto e permite a criação de aplicações para dispositivos móveis baseados em iOS e Android, além de aplicações para navegadores *Web* e *desktops*. Através da linguagem MXML, que é uma linguagem de marcação baseada em XML, o Flex SDK permite a criação de interfaces, além de fornecer uma vasta coleção de componentes que permitem a construção de aplicações complexas praticamente de forma declarativa. Para tratar de aspectos de interatividade, o Flex utiliza a linguagem ActionScript <sup>3</sup>, que é parte fundamental da plataforma Flash. Deste modo, os códigos gerados pelo CLO Studio na sua versão atual, são compilados e executados como arquivos da plataforma Flash.

A escolha por Flash é justificada pelo fato dessa linguagem ser atualmente, a mais utilizada para a criação de aplicações interativas na *Web*, visto que, segundo estatísticas de julho de 2011, o Adobe *Flash Player* conseguiu atingir 99% do mercado de PCs (SYSTEMS, 2011). Além disso, por ter sido umas das primeiras linguagens de desenvolvimento interativo da Internet, uma grande quantidade de OA disponíveis nos repositórios no mundo na atualidade são desenvolvidos nessa linguagem (HOLZINGER; EBNER, 2003)(HARMAN; KOOHANG, 2007).

É importante ressaltar que a escolha da plataforma Flex/Flash não impõe nenhuma restrição ao desenvolvimento do CLO Studio, sendo essa DSL completamente independente de plataforma alvo. Sendo assim, as abstrações tratadas no CLO *Studio* são passíveis de geração de código para qualquer linguagem que utilize o conceito de elementos visuais e eventos de controle de interação. Desta forma, para que o CLO *Studio* possa gerar código para linguagens como HTML 5 (HICKSON; HYATT, 2008), que está despontando atualmente como o maior concorrente e um possível substituto do Flash no futuro, basta apenas que os *scripts* de geração de código sejam reescritos para essa linguagem, sem haver a necessidade de realizar qualquer modificação na implementação do CLO Studio como também de qualquer aplicação construída com essa DSL. Assim, para que um OAC construído com o CLO *Studio* possa ser transformado em HTML 5, basta incluir o *script* de transformação no CLO *Studio* e solicitar a geração

---

<sup>2</sup><http://www.adobe.com/br/products/flex.html>

<sup>3</sup>[http://help.adobe.com/en\\_US/as3/dev/index.html](http://help.adobe.com/en_US/as3/dev/index.html)



do novo código.

### 5.2.3 Ferramentas Utilizadas

O fato do CLO *Studio* ser um *plugin* do Eclipse, possibilita a utilização de diversas outras ferramentas e *plugins* disponíveis para essa plataforma. No Quadro 5.1 estão listadas todas as ferramentas e outros *plugins* utilizados na produção do CLO *Studio*.

Quadro 5.1: Ferramentas Utilizadas no Desenvolvimento do CLO Studio

Ferramenta	Descrição	Utilidade
GMF <sup>a</sup>	<i>Framework</i> do Eclipse que faz parte do GMP ( <i>Graphical Modeling Project</i> ) e é utilizado para a geração de editores gráficos baseados em EMF ( <i>Eclipse Modeling Framework</i> ) e no GEF ( <i>Graphical Editing Framework</i> )	Esse <i>framework</i> é a base de implementação do CLO Studio. Através dele toda a estrutura dessa ferramenta foi construída.
Eugenia <sup>b</sup>	É uma ferramenta utilizada para gerar alguns dos modelos intermediários do GMF a partir de anotações nos arquivos Ecore	Essa ferramenta deu suporte à construção do metamodelo do CLO Studio.
Emfatic <sup>c</sup>	É uma notação textual para a criação de documentos Ecore	Essa abordagem textual foi utilizada para a descrição do metamodelo do CLO Studio.
EVL <sup>d</sup> ( <i>Epsilon Validation Language</i> ) e EOL <sup>e</sup> ( <i>Epsilon Object Language</i> )	Ambas as linguagens fazem parte do projeto Epsilon e realizam, respectivamente, a descrição de restrições em metamodelos EMF e a manipulação programática desses arquivos	Essas linguagens foram utilizadas para a criação das regras de validação dos modelos do CLO Studio

Ferramenta	Descrição	Utilidade
Apache FOP <sup>f</sup> ( <i>Formatting Objects Process</i> )	É uma API java que faz a leitura de uma árvore de objetos e realiza a geração de documentos em diversos formatos, tais como <i>Portable Document Format</i> (PDF), <i>Extensible Markup Language</i> (XML) e <i>Portable Network Graphics</i> (PNG)	Essa API foi utilizada para a criação dos documentos gerados pelo CLO Studio, como o Roteiro de Atividades ( <i>storyboard</i> )
Flex SDK <sup>g</sup>	É um <i>framework</i> de código aberto para a implementação de aplicações em Flex	Esse <i>framework</i> foi utilizado para a geração de código Flash pelo CLO Studio. Esses códigos são utilizados tanto para a simulação das aplicações como para a criação da versão final do OAC
Apache Ant <sup>h</sup>	É uma biblioteca que permite a criação de <i>scripts</i> de compilação de aplicações	Foi utilizada na geração de todos os arquivos compilados do CLO Studio
Senocular Transformation Tool <sup>i</sup>	É uma ferramenta desenvolvida em <i>Actionscript</i> utilizada para a realização de transformações em componentes visuais desenvolvidos em <i>Actionscript</i> .	Toda a manipulação dos elementos visuais, tais como mover e redimensionar, disponíveis na customização de um OAC, foi desenvolvido através dessa ferramenta
Mofscript <sup>j</sup>	É uma linguagem de geração de textos a partir de modelos. Essa linguagem é independente de metamodelos específicos	A geração de todos os códigos do CLO <i>Studio</i> foi realizada utilizando essa linguagem

Fonte: Própria Autora

<sup>a</sup>O GMF pode ser baixado juntamente com o *Eclipse Modeling Tools* (<http://www.eclipse.org/downloads/>)

<sup>b</sup><http://eclipse.org/gmt/epsilon/doc/eugenia/>

<sup>c</sup><http://wiki.eclipse.org/Emfatic>

<sup>d</sup><http://eclipse.org/gmt/epsilon/doc/ev1/>

<sup>e</sup><http://eclipse.org/gmt/epsilon/doc/eol/>

<sup>f</sup><http://xmlgraphics.apache.org/fop/>

<sup>g</sup><http://opensource.adobe.com/wiki/display/flexsdk/Flex+SDK>

<sup>h</sup><http://ant.apache.org/>

<sup>i</sup><http://www.senocular.com/?entry=801>

<sup>j</sup><http://eclipse.org/gmt/mofscript/>

## 5.3 Projeto da Linguagem

O objetivo da linguagem associada ao CLO *Studio* é possibilitar a especificação de OAC do tipo animação/simulação de forma simples e com vocabulário acessível para todos os participantes do processo de desenvolvimento deste tipo de recurso. Para conseguir esse objetivo, foram utilizados elementos de modelagem oriundos desse tipo de OA e de controles de aplicações interativas em geral, tais como, eventos de mouse, eventos de tempo, dentre outros. Esses elementos foram organizados em uma estrutura hierárquica baseada na ideia de máquinas de estados.

Para apresentar todas os detalhes do projeto da linguagem do CLO *Studio*, na Seção 5.3.1 os Conceitos Modelados por essa ferramenta são descritos. Posteriormente, as Regras de Validação para especificações de modelos dessa ferramenta são detalhadas na Seção 5.3.2. Na Seção 5.3.3 é apresentada a Notação visual para os elementos representados pelo CLO *Studio*. Em seguida, na Seção 5.3.4, a interface do CLO *Studio* é descrita. Para finalizar, na Seção 5.3.5 são apresentados alguns Exemplos de modelos produzidos com o CLO *Studio*. Esses exemplos tem por finalidade demonstrar tanto as notações quanto as funcionalidades dessa ferramenta.

Com a intenção de ampliar a visibilidade do CLO *Studio*, a interface dessa ferramenta foi implementada em língua inglesa. Deste modo, daqui em diante, quando os elementos da interface do CLO *Studio* forem citados no texto eles aparecerão tanto em português como em inglês.

### 5.3.1 Conceitos Modelados

Os conceitos modelados pelo CLO *Studio* podem ser divididos em quatro categorias gerais: Cenas (*Scenes*), Controles (*Controls*), Mídias (*Medias*) e Ações (*Actions*).

- Cenas (*Scenes*): identificam as interfaces do OAC. Cada cena pode ser constituída por um conjunto de outros componentes, tais como mídias e controles, como também pode ser compostas por outras cenas, neste caso, não pode haver outros componentes dentro da cena pai, sendo que esses ficam dentro de subcenas. As cenas de um OAC devem ser interpretadas como telas do OAC. Neste caso as subcenas devem ser vistas como sendo telas que têm seus componentes modificados em tempo de execução sem, no entanto, haver uma transição para outra tela. Uma cena importante no contexto do CLO Studio é a que representa o ponto de início da execução de um OAC. Essa cena, denominada de “*start*”, indica que a cena apontada por ela é a primeira a ser apresentada.
- Controles (*Controls*): devem ser utilizados como mecanismos de acesso às ações em um OAC. Esses controles são apresentados como componentes de interface, tais como botões, imagens e textos. Já se estuda a possibilidade de outros controles serem incluídos no CLO *Studio*, como por exemplo *combobox*, *listbox*, dentre outros
- Mídias (*Medias*): são apresentadas como componentes que podem ser “tocados” pelo CLO *Studio*, tais como vídeo e áudio.
- Ações (*Actions*): essas ações indicam a transição entre cenas. Essas transições podem ocorrer por ações de *mouse* (denominadas “*mouse actions*”), como um clique ou a passagem do *mouse* sobre um componente do OAC; por ações temporizadas (denominadas “*time actions*”), tais como a apresentação de uma cena ao se passar determinado tempo; ou como a indicação da ação de execução da primeira cena de um OAC (denominadas “*start actions*”).

Além da descrição das cenas, de seus componentes e ações, também podem ser especificados no CLO *Studio* as Ações de Customização associadas a todos esses tipos de componentes. As ações de customização, como apresentado no Capítulo 4,

estão relacionadas a componentes específicos do OAC. Deste modo, essas ações foram implementadas como sendo propriedades de cada componente. Assim, a ação de “substituir”, por exemplo, é implementada como sendo uma propriedade do componente do tipo “Imagem” que pode ser ativada por quem estiver especificando o OAC, indicando que esse componente poderá ser substituído quando da execução do OAC.

Para ilustrar a relação entre todos os componentes do CLO *Studio*, a Figura 5.3 apresenta a especificação do metamodelo no formato ECore. No Apêndice A pode ser encontrada a versão desse metamodelo descrita na linguagem Emfatic com as anotações do Eugenia. A utilização do Emfatic no CLO *Studio* permitiu a construção de especificações textuais com a possibilidade da inserção de anotações do Eugenia. Com essas anotações foi possível a otimização do processo de geração dos códigos pelo Eclipse/GMF, visto que com o Eugenia<sup>4</sup>, várias etapas intermediárias puderam ser desconsideradas.

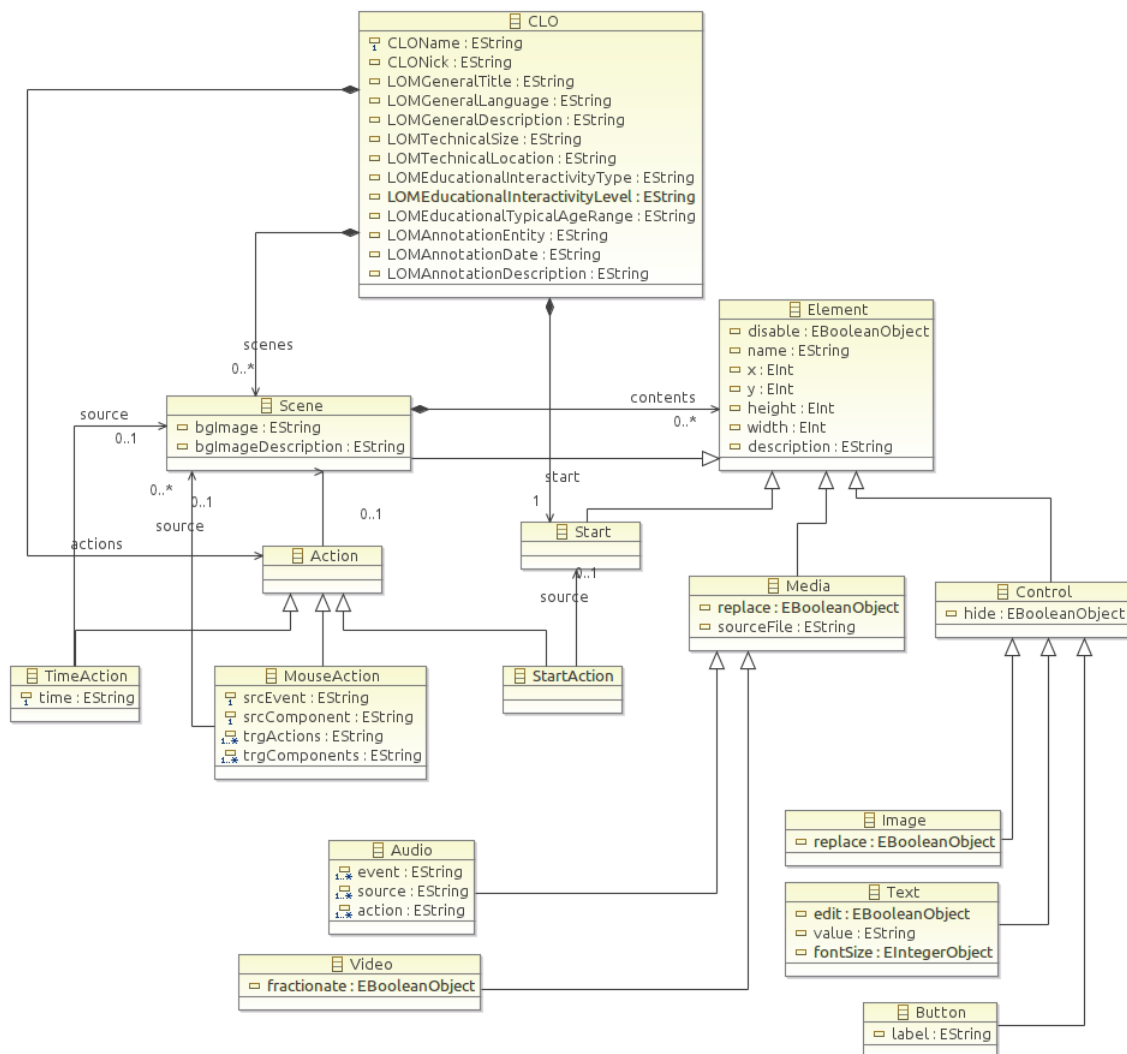
Os elementos apresentados no metamodelo da Figura 5.3 estão organizados de forma hierárquica, tendo o elemento CLO como a raiz do metamodelo. Esse elemento possui diversas propriedades que servem para identificar o modelo a ser construído (*name* e *nick*) e outras que estão relacionadas aos metadados LOM (*LOMGeneralTitle*, *LOMGeneralLanguage*) que são utilizados para registrar os OAC em repositórios que utilizem esse padrão. Além do elemento CLO, o metamodelo do CLO *Studio* possui os seguintes outros elementos:

- ELEMENT: representação geral dos elementos do metamodelo. Dele deriva todos os demais elementos. Suas propriedades estão relacionadas à identificação dos elementos (*name* e *description*) e ao posicionamento e dimensionamento dos mesmos nas cenas (*x*, *y*, *width* e *height*).
- SCENE: representa uma cena. Suas propriedades privadas (*bgImage* e *bgImage-Description*) permitem a definição do *background* da cena. Além de ser derivada de ELEMENT, essa classe também é composta por um conjunto de elementos do tipo ELEMENT, fato que indica a possibilidade de haver cenas aninhadas, o que representa o conceito de subcena.

---

<sup>4</sup><http://www.eclipse.org/gmt/epsilon/doc/articles/eugenia-gmf-tutorial/>

Figura 5.3: Metamodelo ECore do CLO Studio



Fonte: Própria Autora

- **START**: esse elemento representa um tipo de cena especial que indica o ponto de início de execução do OAC.
- **MEDIA**: classe que incorpora os elementos **AUDIO** e **VIDEO**. Suas propriedades gerais são *replace*, que é uma variável lógica que permite a definição de mídias que podem ser substituídas e *sourceFile*, que indica o nome do arquivo associado à mídia. Já a classe **AUDIO** possui as propriedades *event*, que indica os eventos que podem ocorrer associados ao áudio; *source* que indica a origem do evento, e *action*, indicando a ação que deve ocorrer em resposta ao evento. A classe **VIDEO** possui a propriedade *fractionate* que é uma variável lógica que permite a definição de vídeos que podem ser temporalmente redimensionados.

- **CONTROLS:** representa os controles da interface de um OAC. Tem *hide* como propriedade, que é uma variável lógica que permite a definição de controles que podem ser escondidos. Tem **IMAGE**, **TEXT** e **BUTTON** como subelementos. O elemento **IMAGE** tem *replace* com propriedade. Essa é uma variável lógica que permite a definição das imagens que podem ser substituídas. **TEXT** tem as propriedades *edit*, que é uma variável lógica que permite a definição de textos que podem ser editados; *value* e *fontSize* que representam, respectivamente, o conteúdo e o tamanho da fonte do texto.
- **ACTION:** representa as ações dos OAC. Pode derivar as classes **MOUSEACTION**, **TIMEACTION** e **STARTACTION**. **MOUSEACTION** indica ações de mouse e tem as propriedades *srcEvent* e *srcComponent*, que indicam, respectivamente, o tipo e o identificador da origem do evento, além de *trgActions* e *trgComponents*, que indicam as ações e os componentes afetados pelo evento. A classe **TIMEACTION** representa ações temporais, possui a propriedade *time*, que indica o tempo que o evento será disparado. A classe **STARTACTION** representa uma ação de inicialização de execução de um OAC, indicando a ligação de um elemento **START** com um elemento **SCENE**.

### 5.3.2 Regras de Validação

As informações contidas no metamodelo da Figura 5.3 conseguem capturar tanto os elementos das interfaces dos OAC como as conexões entre eles. Contudo, essas informações não conseguem cobrir todas as regras de domínio dos OAC. Por esse motivo, regras de validação foram introduzidas de modo a garantir a consistência das especificações dos modelos gerados pelo *CLO Studio*. Nas próximas subseções são descritas as regras implementadas por essa ferramenta de autoria.

#### 5.3.2.1 Regras Gerais

Essas regras tratam tanto da organização da arquitetura de um OAC quanto das informações gerais relacionadas a cada elemento. As seguintes regras gerais estão disponíveis no *CLO Studio*:

- Só deve existir um único componente do tipo “*start*” em um OAC e esse elemento deve apontar para a primeira cena que será apresentada quando da execução do OAC. Caso o OAC tenha uma cena com subcenas também deve ser indicada a subcena principal com a utilização do elemento “*start*”, que também deve ser único dentro de cada cena.
- O nome do OAC não pode estar em branco.
- Todas as cenas devem ter conexões com outras cenas.
- As cenas devem ter nomes diferentes.

### 5.3.2.2 Regras de Elementos

Essas regras tratam da validação de informações dos componentes que estão relacionadas às cenas de um OAC.

- Nenhum componente pode ter um nome em branco.
- Não pode haver espaços em branco nos nomes dos componentes.
- Não pode haver nomes de elementos repetidos dentro de uma mesma cena.
- Cada subcena deve ter o mesmo nome da cena pai.

### 5.3.2.3 Regras de Navegação

As regras de navegação estão ligadas à validação das ações de interatividade e de controle de sequenciamento de eventos em um OAC.

- Não pode haver eventos de mouse sem a completa identificação de origem e destino dos eventos.
- Não pode haver eventos temporais sem a identificação do tempo da transição.



Todas as regras de validação foram implementadas no CLO Studio através de especificações EVL. No Apêndice B essas regras estão devidamente descritas nesta linguagem.

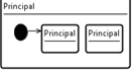




### 5.3.3 Notação




A notação utilizada pelo CLO *Studio* foi desenvolvida em paralelo à construção do metamodelo. Para cada componente foi procurado fornecer uma representação de fácil assimilação tanto por profissionais de *design* e desenvolvedores de software, bem como por professores com experiência na utilização de computadores. Desta forma o Quadro 5.2 apresenta as principais notações utilizadas na representação dos componentes no CLO *Studio*.

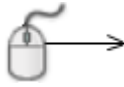


Quadro 5.2: Notações do CLO Studio

Componente	Notação	Descrição	Propriedades Principais
Início ( <i>Start</i> )		Indica o ponto de partida para a execução do OAC. Tanto deve ser usado para indicar a cena principal do OAC como para indicar a subcena principal de uma cena.	Não se aplica
Cena ( <i>Scene</i> )		Representa uma cena cujo nome deve ser especificado na parte superior.	<i>Name</i> : o nome que a cena recebe. <i>Background Image Name</i> : indica o nome de um arquivo com uma imagem (no formato swf) que deve ser utilizado como <i>background</i> da cena. <i>Background Image Description</i> : realiza a descrição da imagem de <i>background</i> requerida.

Componente	Notação	Descrição	Propriedades Principais
Subcenas		<p>Neste exemplo a cena “principal” é formada por duas subcenas. Conforme explicado nas regras de validação, cada subcena deve ter o mesmo nome da cena principal. Também é necessário indicar qual a primeira subcena a ser apresentada. Isso é realizado através da ligação de uma cena “<i>start</i>” através de uma “<i>start action</i>”.</p>	<p>Mesmas das Cenas, exceto <i>Background Image Name</i> e <i>Background Image Description</i></p>

Componente	Notação	Descrição	Propriedades Principais
<p>Áudio (<i>Audio</i>)</p>		<p>Arquivos de áudio são representados por este componente.</p> <p>Atualmente apenas arquivos no formato mp3 são aceitos como fontes de áudio do CLO Studio.</p>	<p><i>Name</i>: nome do componente.</p> <p><i>Source File</i>: nome do arquivo de áudio. <i>Event</i>: evento que deve ser “disparados” pelo controle. Atualmente são implementados eventos “<i>click</i>” e “<i>start</i>”. <i>Source</i>: determina a origem do evento especificado anteriormente.</p> <p><i>Action</i>: indica a ação que deve ser realizada. No caso são implementadas as seguintes ações: <i>play</i>, <i>pause</i> e <i>stop</i>.</p> <p><i>Description</i>: descrição do componente.</p>
<p>Vídeo (<i>Video</i>)</p>		<p>Arquivos de vídeo são representados com esta notação pelo CLO Studio.</p> <p>Atualmente, apenas arquivos no formato “flv” são aceitos.</p>	<p><i>Name</i>: nome do componente.</p> <p><i>Source File</i>: nome do arquivo de vídeo. <i>Description</i>: descrição do componente.</p>

Componente	Notação	Descrição	Propriedades Principais
Imagem ( <i>Image</i> )		<p>As imagens são representadas desta forma no CLO <i>Studio</i>. Atualmente, apenas imagens no formato “swf” são aceitas. Essas imagens podem receber eventos de mouse e, portanto, podem funcionar como botões para o CLO <i>Studio</i>.</p>	<p><i>Name</i>: nome do componente. <i>Description</i>: descrição do componente.</p>
Botão ( <i>Button</i> )		<p>Botões são representados desta forma. Estes componentes representam elementos de interação que podem receber eventos de mouse.</p>	<p><i>Name</i>: nome do componente. <i>Label</i>: legenda apresentada no botão. <i>Description</i>: descrição do componente.</p>
Texto ( <i>Text</i> )		<p>Textos escritos no CLO</p>	<p><i>Name</i>: nome do componente. <i>Value</i>: conteúdo de texto. <i>Font Size</i>: tamanho da fonte do texto. <i>Description</i>: descrição do componente.</p>

Componente	Notação	Descrição	Propriedades Principais
Ação de Mouse ( <i>Mouse Action</i> )		Representa ações iniciadas por eventos de mouse.	<i>Source Component</i> : componente de origem do evento. <i>Source Event</i> : tipo de evento (atualmente implementados <i>Click</i> e <i>MouseOver</i> ). <i>Target Components</i> : lista de componentes afetados com o evento. <i>Target Actions</i> : lista de ações para cada componente de destino. Atualmente, são implementadas ações de <i>VisibleOn</i> , <i>VisibleOff</i> , <i>SwitchVisible</i> e <i>show</i> .
Ação de Tempo ( <i>Time Action</i> )		Representa ações temporais.	<i>Source</i> : indica o componente origem do evento temporal. <i>Time</i> : indica o tempo a ser utilizado para disparar o evento.
Ação de Início ( <i>Start Action</i> )		Representa ação de <i>start</i> . Apenas elementos do tipo “ <i>Start</i> ” podem ser utilizados como origem desse tipo de evento.	Não se aplica.

Fonte: Própria Autora

Além dos componentes listados no Quadro 5.2, algumas outras propriedades podem ser destacadas na descrição de um OAC. Essas propriedades dizem respeito ao projeto do OAC como um todo e não a componentes específicos da interface do mesmo. Estas,

por sua vez, foram divididas em duas categorias: General e LOM<sup>5</sup>. A primeira categoria possui apenas uma propriedade na qual é indicado o nome do OAC a ser desenvolvido no projeto. A segunda categoria é utilizada para descrever os metadados no formato *Learning Object Metadata* (LOM) do recurso atual. Essa categoria, por sua vez, é dividida em diversas propriedades, cobrindo algumas das seções identificadas no padrão LOM. É importante observar que essa ainda não é a definição de um perfil de aplicação LOM<sup>6</sup> de aplicação para um OAC, mas apenas uma versão inicial de um mecanismo padronizado de catalogação deste tipo de recurso. No Quadro 5.3 são apresentadas as principais categorias e elementos do LOM utilizados no CLO *Studio*. Esses elementos foram selecionados por tratarem diretamente de informações relacionadas a aplicações do tipo simulação/animação que são implementadas pelos OAC.

Quadro 5.3: Metadados LOM dos OAC

<b>Categoria</b>	<b>Elemento</b>	<b>Descrição</b>
<i>General</i>	Title	Nome dado ao OAC. Esse valor pode ser preenchido tanto nos metadados LOM como diretamente na propriedade “ <i>Name</i> ” do OAC.
	<i>Language</i>	Linguagem do OAC.
	<i>Description</i>	Descrição textual do conteúdo do OAC.
<i>Technical</i>	<i>Size</i>	Tamanho do recurso em bytes.
	<i>Location</i>	<i>String</i> utilizada para acessar o OAC. Pode ser uma URL.
<i>Educational</i>	<i>Interactivity Type</i>	Modo de aprendizagem predominante para o recurso. No caso de um OAC criado pelo CLO <i>Studio</i> , o valor padrão atualmente é “ <i>active</i> ”.
	<i>Interactivity Level</i>	O grau de interatividade do recurso. No caso de um OAC criado pelo CLO <i>Studio</i> , o valor padrão atualmente é “ <i>high</i> ”.
	<i>Typical Age Range</i>	Faixa de idade prevista para o recurso.
<i>Annotation</i>	<i>Entity</i>	Identificação da pessoa ou entidade que criou a anotação.
	<i>Date</i>	Data da criação da anotação.
	<i>Description</i>	Valor da anotação.

Fonte: Própria Autora

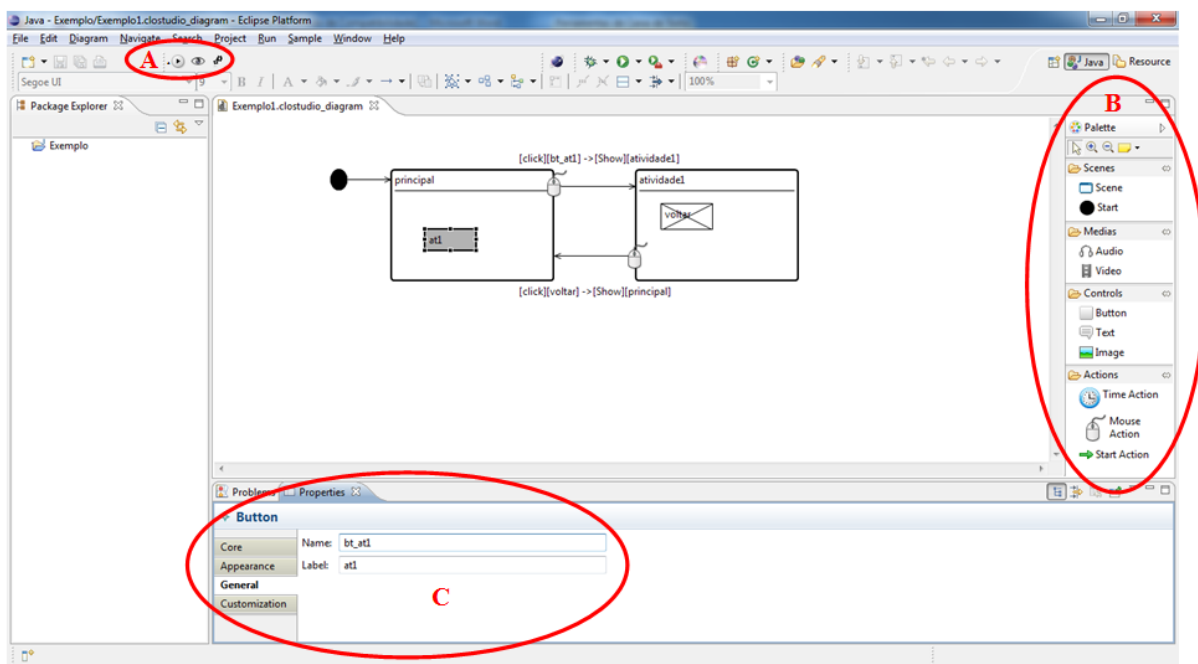
<sup>5</sup> LOM (Learning Object Metadata) é um padrão aberto proposto pelo IEEE LTSC para a catalogação de Objetos de Aprendizagem.

<sup>6</sup>Um perfil de aplicação LOM (*LOM Application Profile*) é uma instância de LOM definida para um tipo de aplicação específica.

### 5.3.4 Interface do CLO Studio

Como um *plugin* do Eclipse, O *CLO Studio* utiliza todas as funcionalidades disponíveis por essa plataforma e inclui alguns itens específicos. Na Figura 5.4 estão destacados os itens principais associados ao CLO Studio na perspectiva(D’ANJOU, 2005) do Eclipse.

Figura 5.4: Interface do CLO Studio



Fonte: Própria Autora

- Barra de Ferramentas (A): nesta barra estão dispostos os controles específicos do CLO Studio para a geração de aplicações e realização de transformações. Os seguintes itens estão disponíveis:
  - *Execute* CLO: permite a execução do OAC que está em projeto. Para que esse item esteja disponível, é necessário que os códigos do OAC tenham sido gerados através do item *Generate* CLO Templates. Também é necessário que os componentes visuais tenham sido desenvolvidos pela equipe de *design* e tenham sido colocados no subdiretório “*components*” dentro do diretório no qual os códigos do OAC tenham sido gerados.
  - *Simulate* CLO: permite a simulação do OAC que está em projeto. Esta simulação permite a visualização das ações de navegação no OAC. Nesta

etapa não é necessário a presença dos componentes visuais, visto que esses são substituídos por formas geométricas de modo a “simular” a presença dos mesmos. Entretanto, tanto as ações como a disposição visual desses componentes são as que farão parte do recurso no final.

- *Generate CLO Templates*: realiza a geração dos códigos do OAC. O desenvolvedor deve escolher um diretório no qual os arquivos serão gerados. Neste diretório serão gerados todos os arquivos Flex (extensão MXML) assim como a documentação associada ao OAC, tais como o Roteiro de Atividades (*storyboard*) e o documento de metadados LOM.
- Paleta (B): apresenta os ícones de acesso aos componentes da linguagem. Essa paleta é dividida em quatro partes, a saber:
  - *Scenes*: representando as cenas. Nessa parte estão os itens *Scene* e *Start*.
  - *Medias*: representam as mídias que podem ser manipuladas. Estão disponíveis os itens *Audio* e *Video*.
  - *Controls*: os controles são representados nesta parte. Estão disponíveis nesta versão os itens *Button*, *Text* e *Image*.
  - *Actions*: representando as ações. Estão disponíveis os itens *Time Action*, *Mouse Action* e *Start Action*.
- Abas de Propriedades (C): essas abas estão disponíveis para cada tipo de componente do modelo, apresentando informações específicas desses componentes. Para cada componente estão disponíveis as seguintes abas:
  - *General*: apresentam informações gerais sobre o componente, tais como “*Name*”.
  - *Customization*: apresentam as ações de customização que estão associadas àquele tipo de componente, permitindo que o projetista possa escolher que operações de customização estarão disponíveis no OAC gerado para o usuário. Por exemplo, em um componente “*Text*”, estarão disponíveis, em forma de *CheckBox*, as seguintes operações de customização: “*Can be Disabled*” e “*Can be Edited*”.



### 5.3.5 Modelos Exemplos

Para melhor compreender tanto a notação utilizada no CLO *Studio* como o próprio funcionamento desta ferramenta, seguem dois exemplos simples de modelos construídos com o CLO *Studio*.

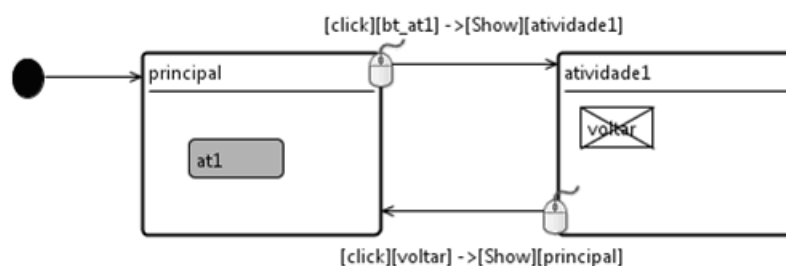
#### 5.3.5.1 Exemplo 1

O primeiro exemplo, apresentado na Figura 5.5, mostra a modelagem da navegação entre duas cenas de um OAC. A execução desse OAC se inicia pela cena “principal”, que possui apenas um botão, cujo nome é “bt\_at1” e cuja legenda é “at1”. A indicação de início é realizada pela presença da cena “start” apontando para “principal” através de uma “start action”.

A segunda cena, denominada “atividade1”, possui apenas um componente imagem, denominado “voltar”.

As duas ações de navegação, ambas do tipo “mouse action”, fazem a conexão entre as duas cenas. A primeira ação, como pode ser visto na legenda superior da conexão na Figura 5.5, é ativada pela execução de um evento “click” no componente “bt\_at1” da cena “principal”. Esse evento leva a execução de uma ação “show” da cena “atividade1”, ou seja, ao clicar no botão “at1” a cena “atividade1” é mostrada.

Figura 5.5: Especificação do OAC do Exemplo 1



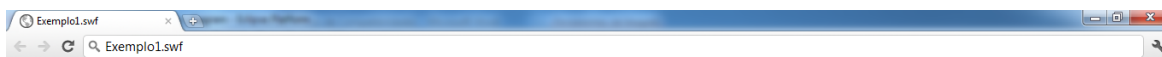
Fonte: Própria Autora

Da mesma forma, a segunda ação é ativada clicando no botão “voltar” da cena “atividade1”, que leva de volta à cena “principal”.

Após as especificações serem realizadas no CLO Studio, é possível proceder uma

simulação do funcionamento do OAC. A Figura 5.6 mostra a cena inicial da simulação, com o botão “at1”. A Figura 5.7, por sua vez, mostra a cena resultado da ação de clicar sobre esse botão.

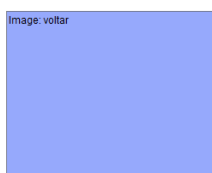
Figura 5.6: Cena inicial da Simulação do Exemplo 1



---

Fonte: Própria Autora

Figura 5.7: Cena resultante da ação de clicar do Exemplo 1



---

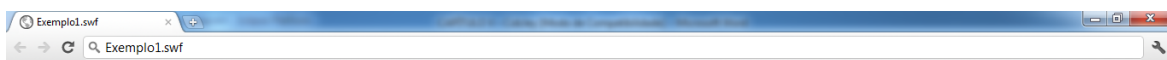
Fonte: Própria Autora

Da mesma forma, ao clicar sobre a imagem na Figura 5.7, será apresentada a cena

principal novamente.

Após verificar o OAC em ação na simulação, o desenvolvedor poderá gerar os códigos (utilizando o botão *Generate CLO Templates* na barra de ferramentas). Com os códigos gerados, a equipe de *design* poderá criar os componentes visuais e salvá-los no subdiretório “components” no diretório no qual o OAC foi gerado. A execução do código final poderá ser visualizada (utilizando o botão *Execute CLO*). As Figuras 5.8 e 5.9 mostram as mesmas telas da simulação, agora com o componente “voltar” devidamente instalado. Nesse exemplo a imagem “voltar” é o único componente que necessita ser implementado pela equipe de *design*.

Figura 5.8: Cena inicial do OAC do Exemplo 1 em execução



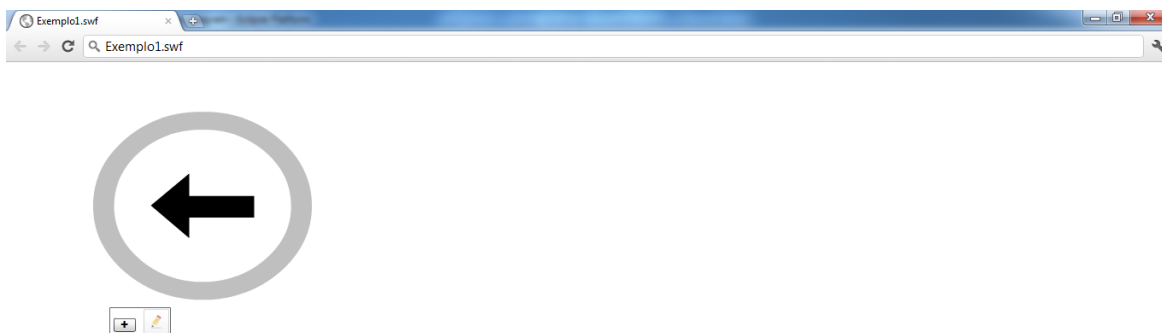
---

Fonte: Própria Autora

### 5.3.5.2 Exemplo 2

O segundo exemplo, apresentado na Figura 5.10, demonstra a utilização do conceito de subcenas do CLO Studio. Este, por sua vez, é uma extensão do Exemplo 1, com a inclusão de subcenas na cena “atividade1”. Nesta cena foram criadas duas subcenas. A primeira, que é indicada pela cena “start”, possui um botão denominado “bt\_mostrar” com legenda “mostrar”. Por ser a primeira subcena desta cena, apenas os componentes pertencentes a esta subcena serão apresentados na tela do navegador. A segunda

Figura 5.9: Cena resultante da ação de clicar no OAC do Exemplo 1 em execução

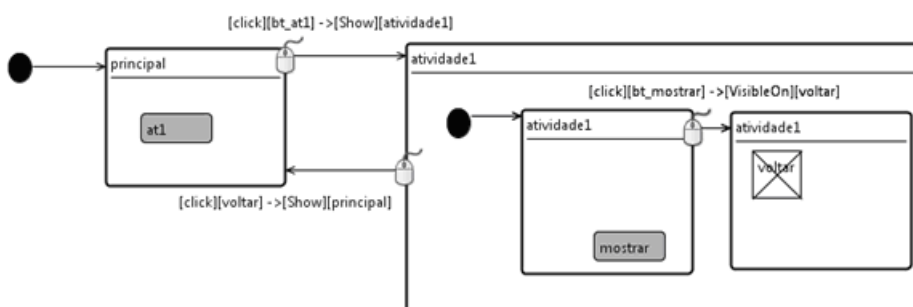


Fonte: Própria Autora

subcena, que contém uma imagem denominada “voltar”, é ativada a partir da realização de um *click* no botão “mostrar”. Esse evento é especificado por uma “*mouse action*” que executa a ação “*visibleOn*” sobre o componente “voltar”. Assim, esse componente será apresentado na mesma cena.

Observando este exemplo, é possível compreender que o conceito de subcena serve para definir elementos que podem ser mostrados ou escondidos a partir de interações do usuário com o OAC. Só após estar visível, o botão “voltar” poderá ser utilizado para retornar à cena “principal”, como especificado pelo “*mouse action*” entre a cena “atividade1” e “principal”.

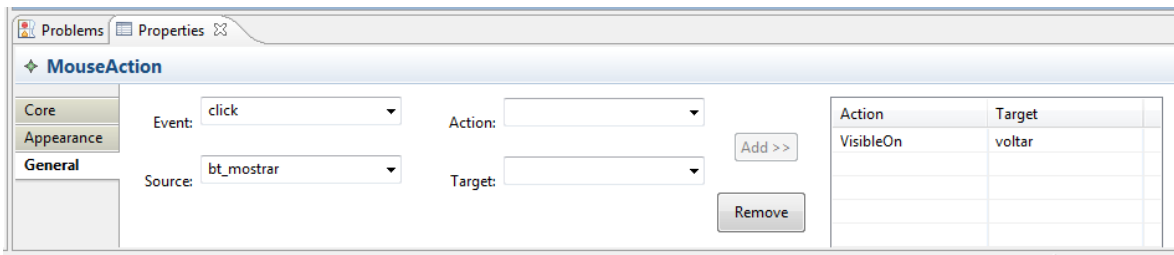
Figura 5.10: Especificação do OAC do Exemplo 2



Fonte: Própria Autora

Na Figura 5.11 é apresentada a forma pela qual a ação associada ao botão “mostrar” é especificada no CLO *Studio*. É possível observar que mais de uma ação pode ser realizada na subcena destino ao se realizar uma ação em algum componente da cena origem. Isso significa que vários elementos podem ser mostrados ou escondidos “simultaneamente” com apenas uma ação de mouse.

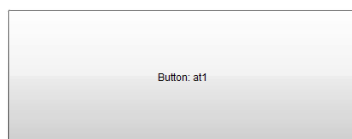
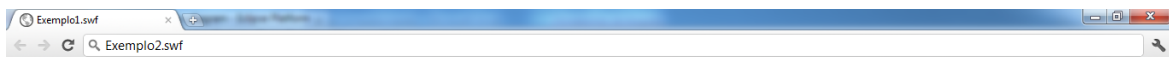
Figura 5.11: Definição de Ação “Mouse Action” no CLO *Studio*



Fonte: Própria Autora

As Figuras 5.12, 5.13 e 5.14 representam, respectivamente, a realização da simulação das ações de interatividade deste OAC. Inicialmente a Figura 5.12 apresenta a cena inicial.

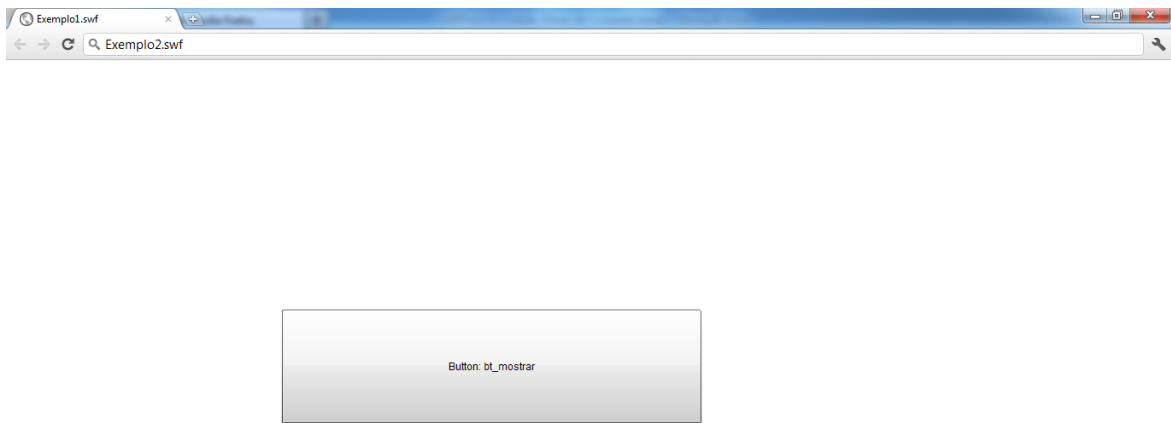
Figura 5.12: Cena inicial da Simulação do Exemplo 2



Fonte: Própria Autora

Ao se clicar no botão “at1” será apresentada a subcena principal da “atividade1”, como mostra a Figura 5.13.

Figura 5.13: Cena resultante da ação no botão “at1” do Exemplo 2

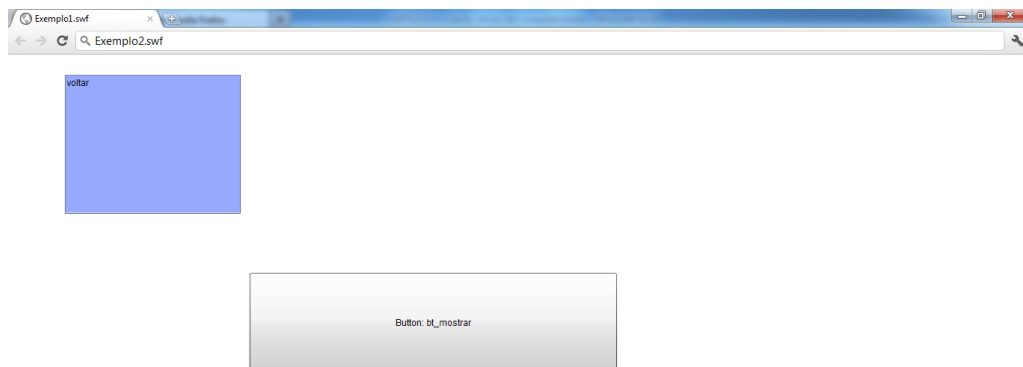


---

Fonte: Própria Autora

Em seguida, ao se clicar no botão “bt\_mostrar”, será visualizada a imagem “voltar” da segunda subcena da “atividade”, como mostra a Figura 5.14. Com o click na imagem “voltar”, o OAC retorna à sua cena inicial.

Figura 5.14: Cena resultante da ação no botão “mostrar” do Exemplo 2



---

Fonte: Própria Autora

Todos os códigos de execução e de simulação são gerados automaticamente pelo CLO *Studio* a partir da descrição do modelo da aplicação. Essa geração ocorre através

da execução de *scripts* de transformação, como apresentado na Seção 5.4.

## 5.4 Transformações

O CLO *Studio* realiza vários tipos de transformações no processo de projeto de um OAC. Algumas dessas transformações, como apresentada no processo de desenvolvimento no Capítulo 4, geram artefatos que são utilizados diretamente pela equipe de desenvolvimento tanto para a implementação do OAC, no caso do Roteiro de Atividades (*storyboard*) e dos Templates de Implementação, como para uma posterior instalação do recurso gerado, como é o caso do arquivo de metadados LOM.

Além dessas, outras transformações ocorrem internamente durante o processo de construção de um OAC. Essas transformações tem a função de gerar os arquivos necessários à simulação do OAC pelo CLO Studio e para a criação do arquivo inicial de controle de versões do OAC. Nas próximas subseções essas transformações são apresentadas em detalhes.

### 5.4.1 Geração dos Templates de Implementação

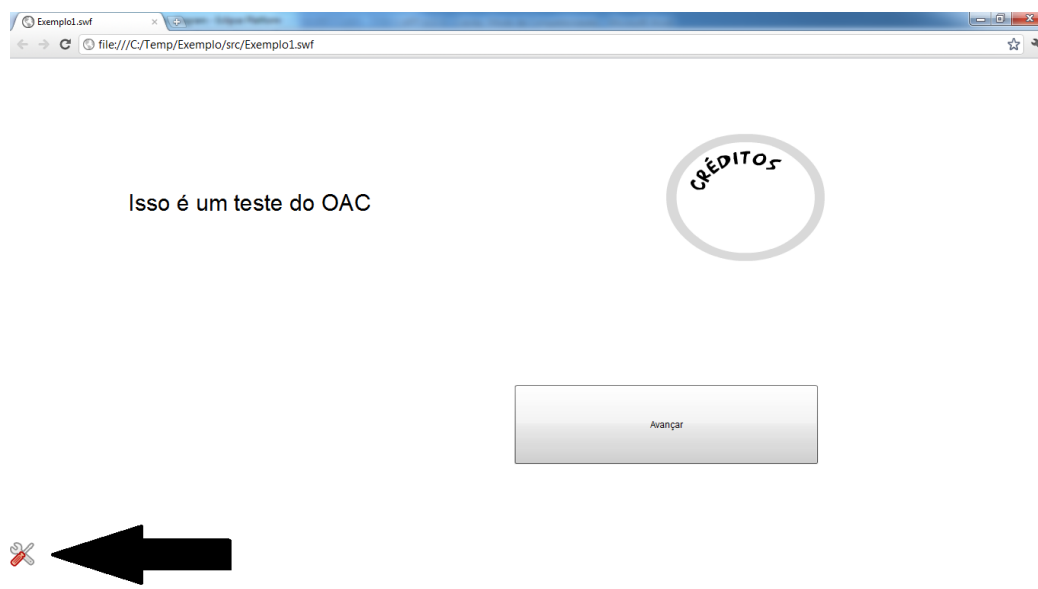
O conceito de *templates* de implementação do CLO *Studio* está ligado aos arquivos gerados na linguagem de programação alvo da ferramenta. Esses *templates* na verdade são códigos completos que podem ser manipulados pelos desenvolvedores para acrescentar e ajustar algum comportamento de um OAC.

Como o CLO *Studio* gera código alvo na linguagem *Flex*, os *templates* de implementação são arquivos no formato MXML que contém os códigos do OAC. Desta forma, a geração desses *templates* está baseada na transformação entre conceitos dos modelos descritos no CLO *Studio* e os componentes da linguagem *Flex*.

Para que se possa entender esse mapeamento e, conseqüentemente, os códigos gerados, é necessário compreender a estrutura geral de um OAC em termos de telas e suas características visuais. A Figura 5.15 apresenta um OAC logo após ser carregado pelo navegador. É importante observar o ícone de customização no canto inferior esquerdo

do recurso (apontado pela seta). Ao ser clicado é apresentada a Barra de Customização. Nessa barra estão dispostos os controles de acesso às ações de customização do referido OAC.

Figura 5.15: Tela de um OAC em Execução



Fonte: Própria Autora

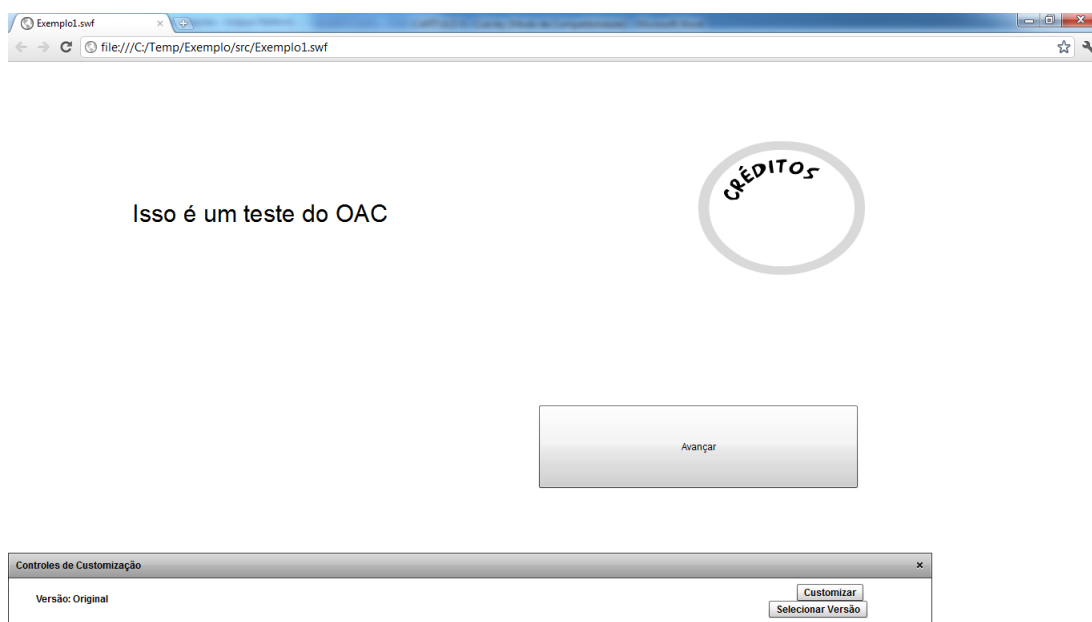
A Figura 5.16 apresenta a Barra de Customização. Através dela, os usuários podem se logar para customizar um OAC (utilizando o botão “Customizar”), ou escolher uma versão já customizada anteriormente (utilizando o botão “Selecionar Versão”).

Para realizar o *login* em um OAC (Figura 5.17), será solicitado um nome de usuário e uma senha. Essas informações devem ser previamente cadastradas em um sistema de controle de acesso ao repositório de OAC. Nesse cadastramento é definido o Grau de Liberdade (GL) do usuário, como apresentado no Capítulo 3. Contudo, esse sistema de controle de acesso não faz parte do escopo desta tese e, por isso, será implementado como um trabalho futuro. Portanto, para fins de avaliação do *CLO Studio*, foi utilizado um arquivo XML com os dados de acesso dos usuários, no qual foram inseridos usuários gerais com *logins* (prof1, prof2, prof3 e prof4), todos com senhas “1234” e, respectivamente, GL 1, 2, 3 e 4.

Após um *login* bem sucedido são apresentadas as Barras de Ações de Customização. Na Figura 5.18, essas barras são mostradas em todos os componentes da interface do OAC. Essa barra é apresentada inicialmente de forma recolhida, como apresentado no

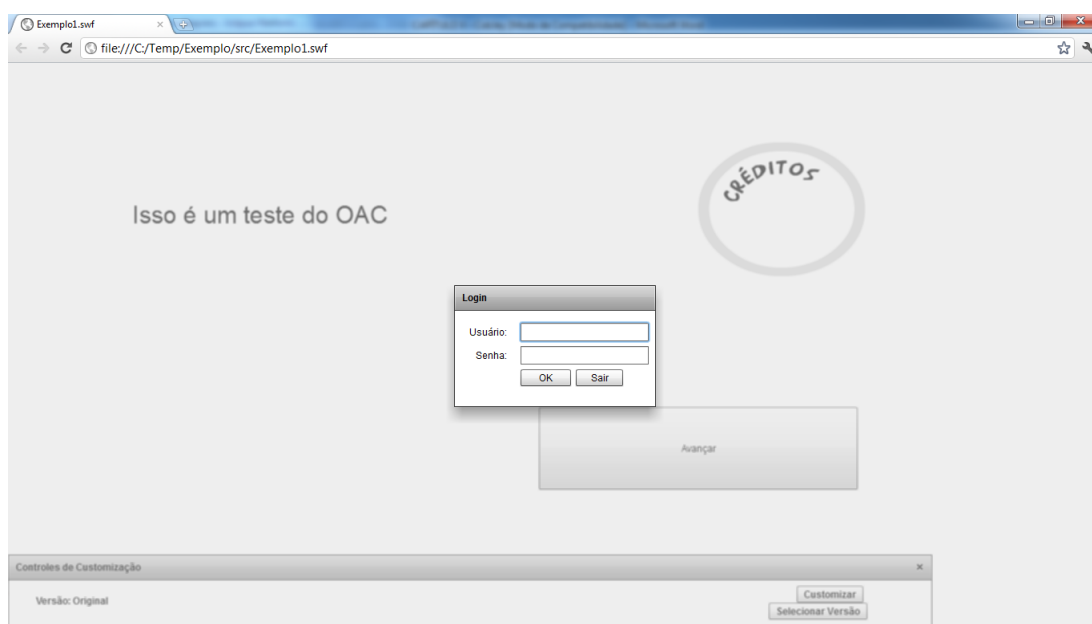


Figura 5.16: Barra de Customização de um OAC



Fonte: Própria Autora

Figura 5.17: Tela de Login em um OAC

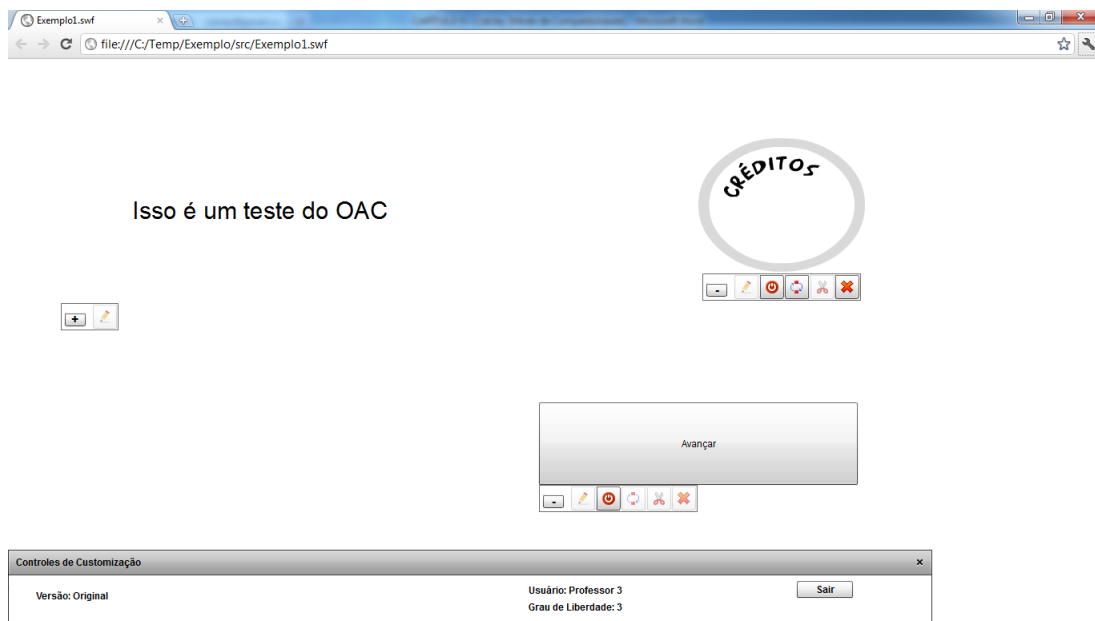


Fonte: Própria Autora

componente do tipo Texto da Figura 5.18. Ela pode ser expandida para mostrar todos os controles de ações de customização, como é o caso dos componentes do tipo Imagem e Botão da Figura 5.18.

Essas barras são flutuantes e podem ser arrastadas pelo usuário. Na Figura 5.19 a barra da imagem “Créditos” está sendo arrastada. Neste momento um conector é

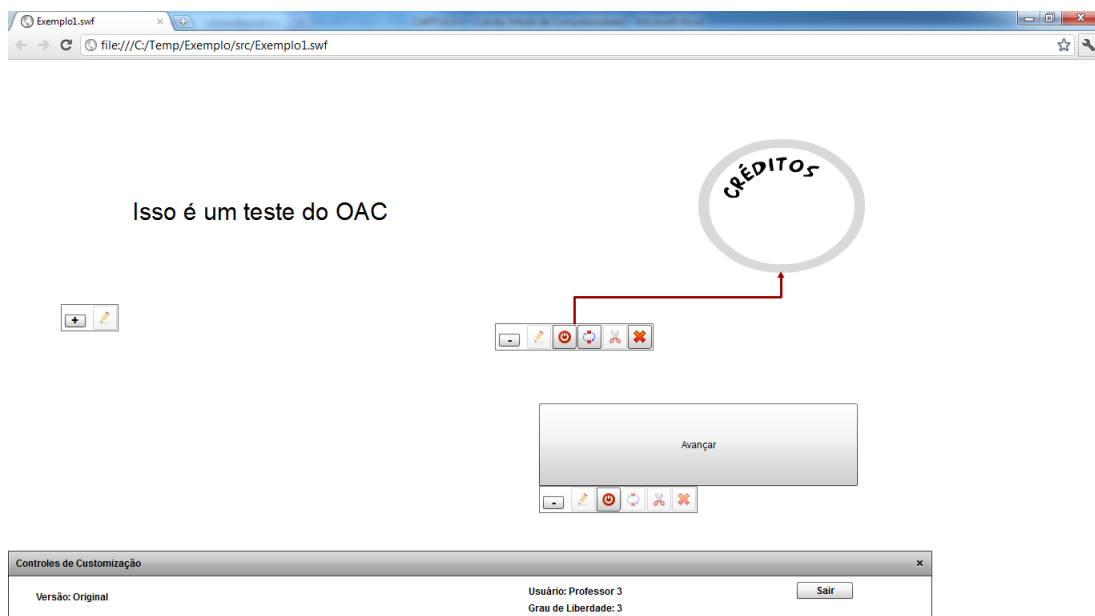
Figura 5.18: Barra de Ações de Customização de um OAC.



Fonte: Própria Autora

apresentado ligando a barra flutuante com o seu respectivo componente. Essa ação facilita a localização dos componentes a serem customizados em uma interface com muitos componentes sujeitos à customização.

Figura 5.19: Conector da Barra de Ações de Customização de um OAC



Fonte: Própria Autora

Nas barras de ações de customização estão dispostos todos os controles de customização, sendo que estão habilitados para uso apenas os controles que foram definidos

pelos desenvolvedores do OAC e que também sejam acessíveis para o GL do usuário que está logado no momento. No Quadro 5.4 são apresentados os elementos que compõem a barra de ações de customização com seus respectivos ícones de acesso.

Quadro 5.4: Elementos da Barra de Ações de Customização

Controle	Ícone
Editar	
Desabilitar/Habilitar	
Substituir	
Fragmentar	
Esconder/Mostrar	

Fonte: Própria Autora

Estabelecida a estrutura visual dos OAC, os códigos gerados devem fornecer de modo automatizado a implementação desses recursos aos usuários. Como explicado anteriormente, os códigos dos OAC gerados pelo CLO *Studio* são construídos a partir do mapeamento de conceitos do metamodelo para a linguagem Flex. Assim, supondo que um OAC seja composto por duas cenas, denominada de “principal” e “atividade1”, os seguintes arquivos são gerados:

- OAC.mxml: arquivo principal da aplicação. Nele são gerados os códigos que inicializam todos os recursos do OAC. Neste caso, esse arquivo é implementado como um componente “*Application*” do *Flex*. Além disso, cada cena do OAC é construído como sendo uma *view* controlada por um componente “*ViewStack*” do *Flex*. Assim, a navegação entre as cenas ocorre pela seleção da respectiva *view* desse “*ViewStack*”.
- Principal.mxml: arquivo com os elementos da cena “principal”.
- Atividade1.mxml: arquivo com os elementos da cena “atividade1”.

Em cada arquivo com a implementação das cenas, os elementos visuais representados no modelo do CLO *Studio* são mapeados em componentes *Flex*. No Quadro 5.5 são apresentados alguns dos mapeamentos que ocorrem diretamente entre esses dois modelos.

Quadro 5.5: Elementos do Metamodelo do CLO *Studio* x Componentes da Linguagem *Flex*

Elemento do Meta-modelo	Componente Flex	Observação
<i>Scene</i>	<i>NavigatorContent</i>	Cada cena do CLO <i>Studio</i> gera um arquivo MXML cujo componente principal é um <i>NavigatorContent</i> . Mesmo as cenas que possuem subcenas geram apenas um arquivo com todos os componentes descritos em todas as subcenas. Assim, uma cena “principal” do CLO <i>Studio</i> , gera um documento com nome “principal.mxml” cujo conteúdo é um <i>NavigatorContent</i> .
<i>Image</i>	<i>Image</i>	Cada imagem gera um componente <i>Flex</i> do tipo <i>Image</i> , tendo como propriedade <i>source</i> o nome do identificador da imagem definido pelo CLO <i>Studio</i> com a extensão “swf”.
<i>Video</i>	<i>VideoDisplay</i> e <i>HSlider</i>	Um vídeo na verdade é uma estrutura mais complexa formada por vários componentes <i>Flex</i> . Os principais são o <i>VideoDisplay</i> , que apresenta o vídeo e o <i>HSlider</i> , que controla o andamento da execução do mesmo.
<i>Audio</i>	<i>Image</i>	Como um áudio é um componente não visual que deve ser manipulado pelo usuário, é criada uma imagem na interface de forma que o usuário possa realizar as customizações necessárias neste componente.
<i>Text</i>	<i>TextArea</i>	Um <i>Text</i> é mapeado para um componente <i>TextArea</i> do <i>Flex</i> .
<i>Button</i>	<i>Button</i>	Um <i>Button</i> é diretamente mapeado em um componente <i>Button</i> do <i>Flex</i> .

Fonte: Própria Autora

Os demais elementos do metamodelo, tais como `TIMEACTION` e `MOUSEACTION`, não possuem um componente correspondente no *Flex*. Esses elementos são mapeados em códigos com funções específicas.

Para gerar um `MOUSEACTION` são consideradas as propriedades desse elemento no metamodelo do CLO *Studio*. Assim, dentro do arquivo `OAC.mxml` é gerada uma função *Actionscript*, denominada `setar_eventos_mouse_action` com a finalidade de inserir “*listeners*” para todos os eventos deste tipo. Assim, para cada evento é incluída uma linha neste método com o seguinte formato:

```
cena_origem.componente_origem.addEventListener  
    (MouseEvent.tipo_evento ,  
     evento_cena_origem_componente_origem);
```

Dessa forma, o seguinte código é gerado para um evento de click no botão “bt\_at1” da cena “principal”.

```
principal.bt_at1.addEventListener(MouseEvent.CLICK,  
    evento_principal_bt_at1);
```

É importante notar que também a função que trata o evento deve ser gerada com as informações do metamodelo. Desta forma, uma função *Actionscript* é gerada no final do arquivo OAC.mxml para cada função definida nos *listeners*. No exemplo anterior, supondo que a ação implementada ao se clicar no botão “bt\_at1” seja a apresentação da cena “atividade1”, a seguinte função deverá ser gerada:

```
public function evento_principal_bt_at1(e:MouseEvent):void{  
    app.selectedChild=atividade1;  
}
```

No caso, app é o nome padronizado do “*StackView*” que controla a visualização das cenas de um OAC.

Para a geração de código dos eventos do tipo TIMEACTION, inicialmente, é gerada uma função, denominada *TimeEvents*, com a seguinte estrutura geral:

```
private function TimeEvents(comp:Object):void{
    if(comp.id=="cena_origem"){
        timer = new Timer(tempo_em_milisegundos,1);
        timer.addEventListener(TimerEvent.TIMER,
                               evento_timer_cena_origem);
        timer.start();
    }
}
```

No caso da ação do exemplo anterior, caso a transição entre a cena “principal” e “atividade1” fosse implementada pela ação temporal de 10s, o seguinte código seria gerado:

```
private function TimeEvents(comp:Object):void{
    if(comp.id=="principal"){
        timer = new Timer(10000,1);
        timer.addEventListener(TimerEvent.TIMER,
                               evento_timer_principal);
        timer.start();
    }
}
```

Da mesma forma que o MOUSEACTION, o TIMEACTION necessita da implementação de uma função para tratar o evento temporal. No caso, será gerada a seguinte função em *Actionscript*:

```
private function evento_timer_principal
    (event: TimerEvent): void {
    app.selectedChild=atividade1;
    timer.stop();
}
```

Outra operação realizada na geração dos *templates* de implementação é o controle do acesso aos recursos do OAC através dos Graus de Liberdade (GL) associados aos usuários. Esse acesso é controlado através da definição de estados (*states*) do *Flex*. Assim, foram definidos cinco estados em um OAC: um estado “*display*”, utilizado na apresentação do OAC sem nenhuma ação de customização, ou seja, sem nenhum usuário logado; e os estados “*edit1*, *edit2*, *edit3* e *edit4*”, correspondentes aos GL de 1 a 4 respectivamente. A apresentação dos elementos de customização na interface do OAC é controlada pelo gerenciamento de estados do OAC, através da utilização da propriedade *currentState*.

Essas são algumas das ações realizadas na geração de *templates* de implementação. O Algoritmo 5.1 apresenta os passos utilizados na geração dos *templates* dos OAC. No *CLO Studio*, essas ações ocorrem através da execução de um *script* implementado na linguagem *Mofscript*.

A função principal do *script* de transformação é a apresentada na linha 1 do Algoritmo 5.1. Nessa função, são invocadas as funções para gerar o arquivo principal da aplicação (linha 2) e para gerar os componentes (linha 3). Posteriormente, esses arquivos são depositados no diretório destino escolhido na execução desse *script* pelo *CLO Studio*. A implementação desse algoritmo em *Mofscript*, realizada no *script* *CLO2Flex.m2t*, é apresentada no Apêndice C.1.

---

**Algoritmo 5.1** Algoritmo de Geração de *Templates* de Implementação do CLO *Studio*

---

1. **função** principal
  2.     aplicação()
  3.     componentes()
  4. **função** aplicação ()
  5.     setar arquivo de saída
  6.     gerar cabeçalho do arquivo
  7.     gerar função init
  8.     carregar arquivo de usuários
  9.     carregar arquivo de versões
  10.    setar cena inicial
  11.    gerar funções
  12.    setar eventos mouseaction
  13.    setar visibilidade subcenas
  14.    setar eventos de áudio
  15.    setar eventos de tempo
  16.    setar estados
  17.    setar viewstack
  18. **função** componentes()
  19.    para cada cena
  20.        setar arquivo de saída
  21.        gerar cabeçalho do arquivo
  22.        setar estados
  23.        gerar tags para componentes da cena
  24.        gerar tags para componentes das subcenas
  25.        gerar elementos de controle de customização
  26.        gerar controles de customização das cenas
  27.        gerar controles de customização das subcenas
- 

### 5.4.2 Geração de Arquivos de Simulação

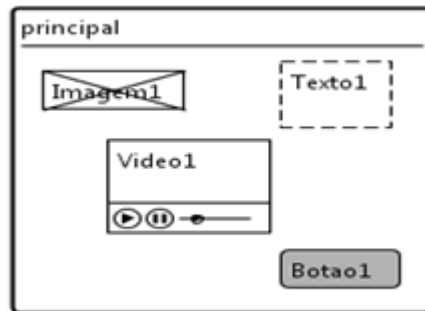
A simulação de uma aplicação pelo CLO *Studio* tem por finalidade possibilitar a visualização prévia tanto da disposição espacial dos componentes na tela como a utilização dos eventos de navegação do OAC. Essa abordagem permite que o OAC possa ser “testado” mesmo sem a implementação dos componentes pela equipe de *design*. Na verdade, a maior utilidade da simulação é permitir que essa atividade de *design* venha a ser realizada sem bloquear o processo de desenvolvimento do recurso.

Dessa forma, para que essa independência entre as fases do processo de desenvolvimento seja alcançada, na simulação, os componentes são substituídos por representações diagramáticas simplificadas. Essas representações são realizadas por retângulos de cores diferentes identificando tipos diferentes de componentes. Por exemplo, a Figura



5.20 mostra o projeto de uma cena no *CLO Studio* na qual são utilizados quatro tipos de componentes, uma imagem (*Imagem1*), um vídeo (*Video1*), um texto (*Texto1*) e um botão (*Botao1*).

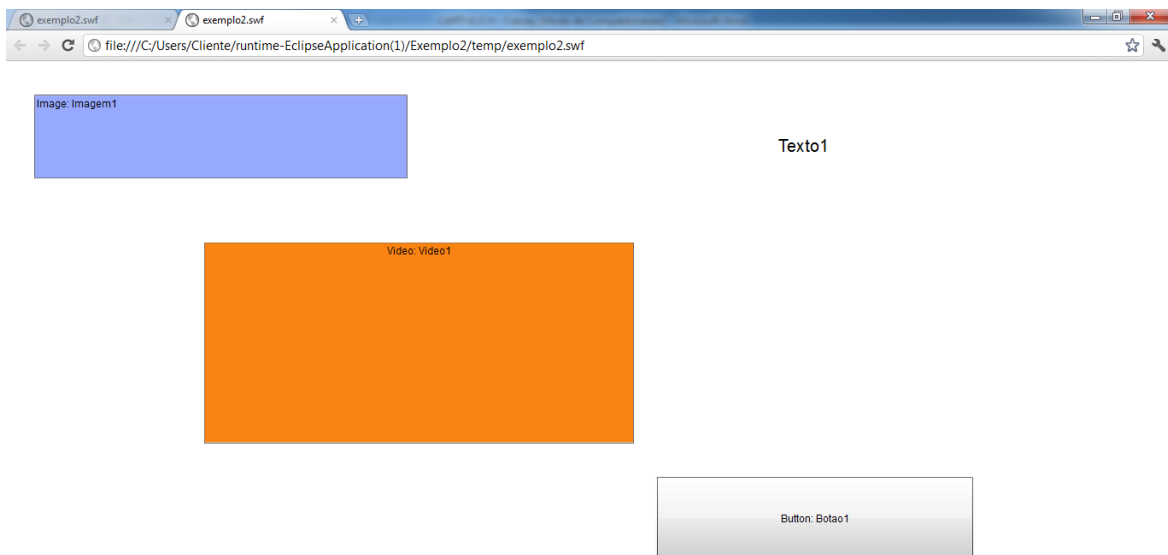
Figura 5.20: Projeto de uma Cena com Vários tipos de Componentes



Fonte: Própria Autora

A Figura 5.21 apresenta a simulação dessa cena no *CLO Studio*. Na simulação, cada tipo de componente é substituído por um retângulo de cor diferente. Em cada um desses retângulos está descrito na sua legenda o tipo de componente e o identificador do mesmo. Assim, no exemplo, o vídeo é representado por um retângulo laranja e tem a legenda “Video: Video1”.

Figura 5.21: Simulação de uma Cena com Vários tipos de Componentes



Fonte: Própria Autora

O algoritmo de geração dos arquivos de simulação é praticamente o mesmo apre-

sentado no Algoritmo 5.1. A diferença básica está na geração dos componentes. Na simulação, as *tags* que invocam imagens e vídeos são substituídas por imagens com cores específicas. Isso permite a execução do OAC sem a necessidade de se ter a priori a implementação desses componentes. Dessa forma, é possível executar o OAC e avaliar a sua estrutura ainda em tempo de projeto. A codificação do algoritmo que realiza essa transformação, implementado no *script* CLO2FlexSim.m2t em Mofscript, é apresentado no Apêndice C.2.

### 5.4.3 Geração do Arquivo de Versões

A customização em um OAC pelos usuários é realizada em tempo de execução. Essa customização tem a finalidade de permitir a construção de diferentes versões do recurso de modo a adaptá-lo às necessidades específicas dos professores. Deste modo, as informações dessas versões devem ser guardadas para que estejam disponíveis para uso futuro. Para realizar o armazenamento persistente das versões dos OAC, o CLO *Studio* utiliza um arquivo XML. Esse arquivo, cuja estrutura básica é apresentada na Figura 5.22, armazena as informações de cada versão do OAC, considerando todas as cenas com todos os seus componentes e valores de atributos associados a esses componentes. Desta forma, quando uma versão é requisitada, esses dados são extraídos do arquivo de versões e os valores dos atributos dos componentes do OAC são modificados para refletir a nova configuração. Da mesma forma, quando uma versão nova é criada, um novo trecho de código é inserido no arquivo XML com as informações da nova versão.

O arquivo de versões da Figura 5.22 apresenta o OAC de nome “Exemplo1” (linha 3). Esse arquivo apresenta apenas uma versão (linha 4 – linha 21), mas outras versões podem ser inseridas (entre as linhas 21 e 23). Também podem ser identificados trechos de duas cenas. A primeira, denominada “principal”, é apresentada entre as linhas 5 e 17. Essa cena é formada por vários componentes. Na Figura 5.22 são apresentados apenas dois componentes (entre as linhas 6 e 10, e entre as linhas 11 e 15).

Para que esse gerenciamento de versões possa ser realizado, é necessário a criação da versão inicial do arquivo de versões. Essa versão inicial representa o OAC sem nenhuma customização, sendo que para a criação de uma nova versão, esses valores

Figura 5.22: Arquivo de Versões Gerado pelo do CLO Studio

```

1. <?xml version="1.0"?>
2. <aplicacoes>
3.     <aplicacao nome="Exemplo1">
4.         <versao val="Original" descricao="Versao Original do OAC.">
5.             <cena nome="principal">
6.                 <componente>
7.                     <nome>texto1</nome>
8.                     <atributo>text</atributo>
9.                     <valor>Isso é um teste do OAC</valor>
10.                </componente>
11.                <componente>
12.                    <nome>bt_credits</nome>
13.                    <atributo>enabled</atributo>
14.                    <valor>>true</valor>
15.                </componente>
16.                ...
17.            </cena>
18.            <cena nome="atividade1">
19.                ...
20.            </cena>
21.        </versao>
22.        ...
23.    </aplicacao>
24.    ...
25. </aplicacoes>

```

Fonte: Própria Autora

são duplicados no arquivo de versões e os valores dos atributos dos componentes são ajustados de acordo com a customização realizada pelo usuário. A criação desse arquivo de versões também é realizada de forma automática pelo CLO Studio através de uma transformação em Mofscript. Essa transformação faz a leitura do modelo e cria o arquivo XML de versões com as informações necessárias. A implementação desse *script* (CLO2Versions.m2t) em *Mofscript* é apresentada no Apêndice C.3.

#### 5.4.4 Geração do Roteiro de Atividade (*Storyboard*)

Como apresentado no Capítulo 4, o Roteiro de Atividades (*storyboard*) é um documento importante utilizado durante o processo de desenvolvimento de um OAC. Ele é gerado tão logo a estrutura do OAC tenha sido discutida e especificada no CLO *Studio*. Com

essa especificação realizada, todas as anotações sobre os componentes a serem construídos pela equipe de *design* são detalhadas e organizadas nesse documento em formato PDF. Assim, cada cena e seus componentes são apresentados na forma tabular, de modo a facilitar a localização dos elementos a serem construídos.

Para a descrição de cada cena, o documento é organizado em três partes. Na primeira parte é apresentada a cena, com sua descrição e uma imagem retirada diretamente do projeto do CLO *Studio*. A segunda parte, que é opcional, descreve as informações de uma imagem que pode ser utilizada como fundo da cena. Na terceira parte, cada componente da cena é detalhado. Nessa descrição são definidos: o nome, o tipo e uma descrição sobre como o componente deve ser construído. É importante ressaltar que o componente a ser construído pela equipe de *design* deve ter o mesmo nome indicado neste documento. A Figura 5.23 apresenta um exemplo de trechos de um *storyboard* gerado pelo CLO *Studio*.

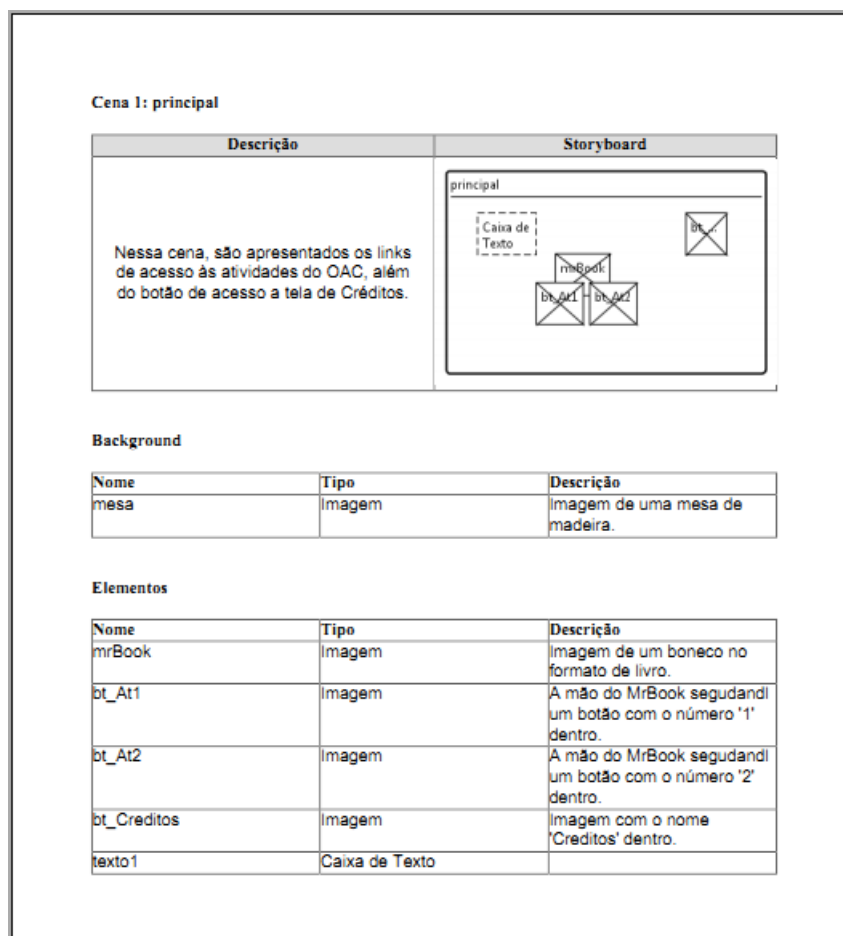
O *script* de geração do Roteiro de Atividades produzido pelo CLO *Studio*, também foi implementado na linguagem *Mofscript*. Esse *script*, como pode ser observado no Algoritmo 5.2, é invocado com um método Java executado pelo CLO *Studio* e que, antes de executar a transformação em si, realiza a extração das imagens de cada cena que está no editor do CLO *Studio* naquele momento e salva cada imagem no diretório no qual o *storyboard* vai ser gerado. Posteriormente, um *script* XSLFO é gerado. Esse *script*, produzido através de uma transformação *Mofscript* (CLO2Fop.m2t), faz referências a cada imagem extraída na etapa anterior (esse *script* é apresentado no Apêndice C.4). Em seguida esse *script* é invocado e o arquivo PDF é gerado (*Storyboard.pdf*), sendo que todos os arquivos temporários são apagados.

---

**Algoritmo 5.2** Algoritmo de Geração de *Storyboards* do CLO *Studio*

---

1. Extrair as imagens de cada cena.
  2. Salvar cada imagem no formato “png”.
  3. Gerar o *script* XSLFO para gerar o arquivo PDF.
  4. Executar o FOP utilizando o *script* gerado na etapa anterior.
  5. Apagar *script* de geração, de compilação e os arquivos de imagem.
-

Figura 5.23: Fragmento de um *Storyboard* Gerado pelo *CLO Studio*

Fonte: Própria Autora

### 5.4.5 Geração de Metadados LOM

Como apresentado na Seção 5.1.2, o *CLO Studio* gera um arquivo de metadados, baseado no padrão LOM, para possibilitar a catalogação dos OAC em repositórios que utilizem esse padrão. Essa operação produz um arquivo XML que segue o padrão do IEEE (IEEE, 2005). A Figura 5.24 apresenta um exemplo de arquivo XML com os metadados LOM gerados pelo *CLO Studio*. As informações contidas neste arquivo foram descritas no Quadro 5.3.

A geração desse arquivo de metadados ocorre também como uma transformação de modelo para texto no qual o *script* de transformação *Mofscript* (*CLO2Lom.m2t*) faz a leitura dos atributos do modelo e produz o arquivo XML com os metadados. Esse *script* é apresentado no Apêndice C.5.

Figura 5.24: Exemplo de Arquivo de Metadados LOM gerado pelo CLO Studio

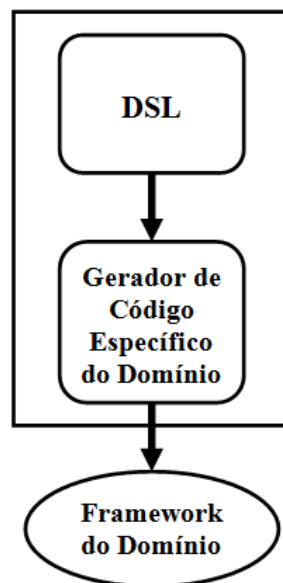
```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <lom xmlns="http://ltsc.ieee.org/xsd/LOM">
3. <general>
4.   <title>
5.     <string language="pt-br">Histórias Fantásticas</string>
6.   </title>
7.   <language>pt-br</language>
8.   <description>
9.     <string language="pt-br">Trabalha produção textual </string>
10.  </description>
11. </general>
12. <technical>
13.   <size>288554</size>
14.   <location>http://www.proativa.vdl.ufc.br/oa/historias/</location>
15. </technical>
16. <educational>
17.   <interactivitytype>
18.     <source>LOM v1.0</source>
19.     <value>active</value>
20.   </interactivitytype>
21.   <interactivitylevel>
22.     <source/>
23.     <value>very high</value>
24.   </interactivitylevel>
25.   <typicalagerange>
26.     <string xml:lang="pt-br">9-10</string>
27.   </typicalagerange>
28. </educational>
29. <annotation>
30.   <entity>
31.     <vcard>Fulano de Tal</vcard>
32.   </entity>
33.   <date>20/01/2010 </date>
34.   <description>
35.     <string xml:lang="pt-br">Esse OA esta completo</string>
36.   </description>
37. </annotation>
38. </lom>
```

Fonte: Própria Autora

## 5.5 *Framework* de Suporte

Como apresentado no início da Seção 5.1, Kelly e Tolvanen (2008) afirmam que um *Framework* de Suporte deve ser fornecido junto com uma DSL com a intenção de facilitar a utilização dessa linguagem. Ainda segundo esses autores, esse *framework* deve ser produzido a partir da percepção de repetições no processo de desenvolvimento, fazendo com que algumas partes desse processo possam ser retiradas como tarefa dos usuários e colocados como uma camada intermediária de software a ser utilizada de modo transparente por quem utiliza a DSL. Dessa forma, a ideia básica é criar uma biblioteca de componentes reusáveis, deixando o usuário apenas referenciar esses componentes. Pelo fato desse *framework* ser desenvolvido com a intenção de dar suporte a uma linguagem específica de um domínio de aplicação, ele também é denominado de *Framework* de Domínio (KELLY; TOLVANEN, 2008). A Figura 5.25 apresenta um *framework* de domínio no contexto de desenvolvimento de uma DSL.

Figura 5.25: *Framework* de Suporte no Contexto de uma DSL



Fonte: Adaptado de Tolvanen (2011)

Para o CLO *Studio* também foi definido um *framework* de suporte com a finalidade de fornecer características mais complexas sem, no entanto, requerer intervenção do usuário final. Para esse fim, foram considerados dois objetivos principais na construção desse *framework*, como apresentado em Kelly e Tolvanen (2008): remover duplicações nos códigos gerados e fornecer funcionalidades adicionais aos usuários finais

da ferramenta.

### 5.5.1 Duplicação de Códigos Gerados

Como observado na geração de código dos OAC, cada cena produz um arquivo MXML separado. Essa fragmentação do código facilita o gerenciamento da aplicação como um todo pela equipe de desenvolvimento. Contudo, diversas ações que são compartilhadas nesses arquivos formam um grande volume de informações redundantes nos códigos gerados. Dessa forma, para “enxugar” os arquivos gerados pelo *CLO Studio* foi criada uma biblioteca de funções implementadas em *Actionscript*. Essa biblioteca, denominada *FuncoesOAC.as*, é importada automaticamente em todos os arquivos gerados pelo *CLO Studio*. Com isso, diversas ações realizadas nas cenas não precisam ser implementadas pelos desenvolvedores. Para exemplificar, esse arquivo realiza ações do tipo: carregar arquivos de usuários e de versões, posicionar elementos nas telas, controlar execução de elementos de vídeo e áudio, dentre outras.

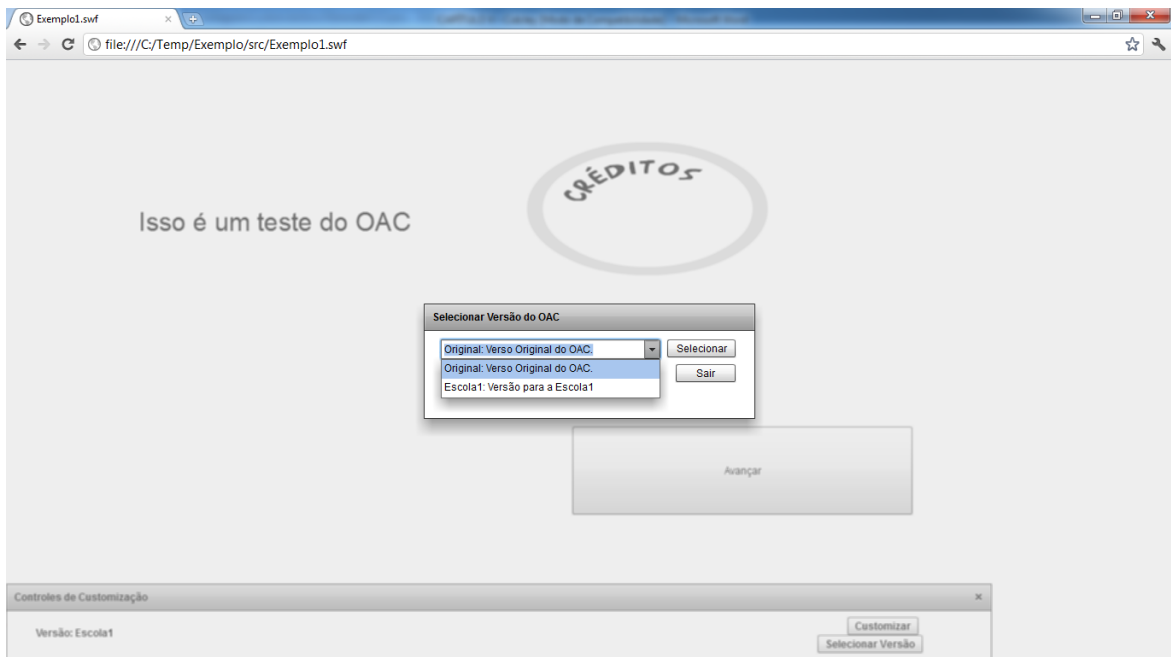
Além dessa biblioteca de funções, outras ações duplicadas foram identificadas no processo de desenvolvimento do *CLO Studio*. Essas ações também foram incorporadas ao *framework* de suporte dessa ferramenta. Como por exemplo a realização das ações de selecionar uma nova versão do OAC, de salvar uma versão que está sendo customizada, a realização da edição de componentes do tipo texto e a fragmentação de vídeos.

A seleção de uma nova versão de um OAC é realizada pelo lançamento de uma janela *pop-up* cuja função é apresentar ao usuário todas as versões daquele OAC disponíveis atualmente no repositório, com a finalidade de permitir a seleção de uma versão nova para ser apresentada no navegador. A Figura 5.26 apresenta essa janela. É possível observar a presença de duas versões do recurso: a versão original e uma versão denominada “Escola1”, já customizada anteriormente.

A ação de salvar uma nova versão de um OAC também é um recurso do *framework* de suporte implementado com uma janela *pop-up*. A Figura 5.27 mostra um exemplo no qual está sendo solicitado o salvamento de um OAC. Nessa janela é inicialmente fornecido como sugestão de nome para o recurso, o nome do OAC que está sendo modificado no momento. Caso esse não seja o OAC original (não é permitido sobrescrever



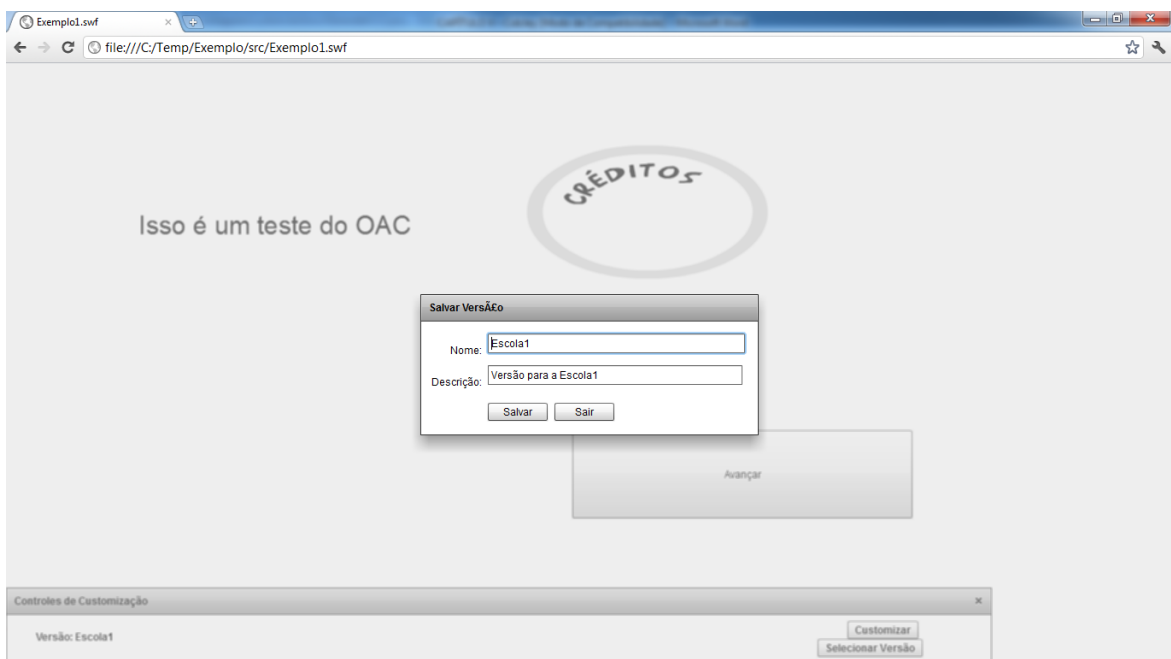
Figura 5.26: Janela de Seleção de Versão de OAC



Fonte: Própria Autora

o OAC original) será solicitada a sobreposição da versão. Caso seja atribuído um nome ainda não disponível no repositório, o OAC é salvo com esse novo nome.

Figura 5.27: Janela de Salvamento da Versão de OAC

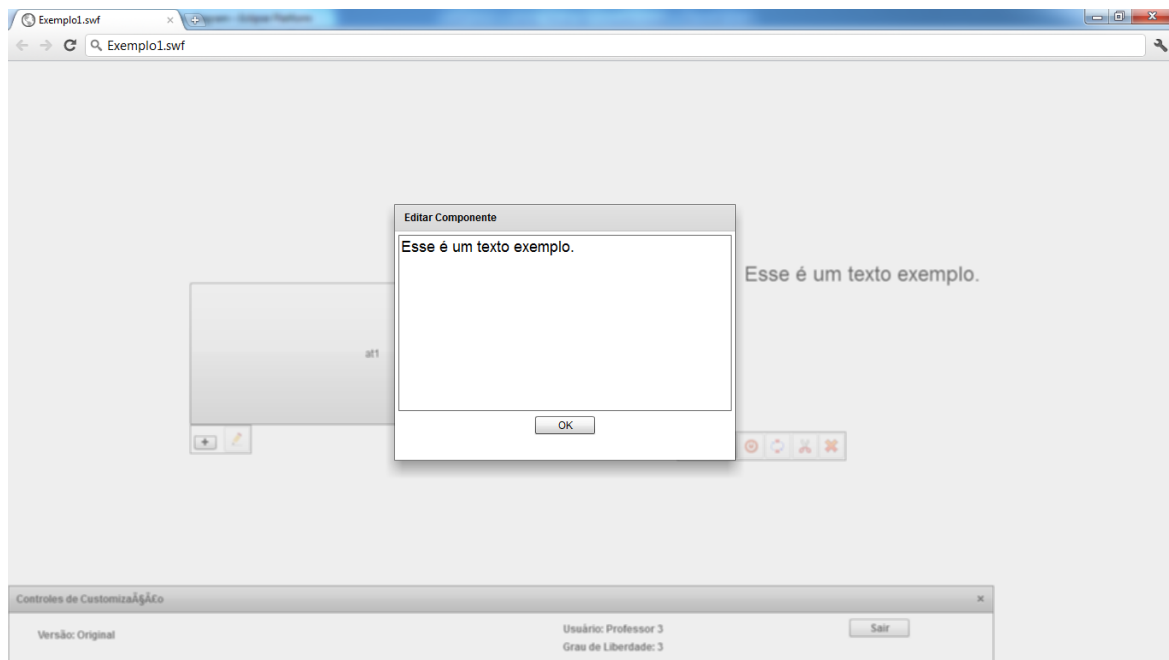


Fonte: Própria Autora

Da mesma forma que as ações anteriores, a edição de um componente texto é

realizado através de uma janela *pop-up*. A Figura 5.28 apresenta um exemplo dessa janela. Nela é possível realizar a edição do conteúdo de um componente texto de um OAC. Ao ser selecionada, essa janela sempre mostrará o conteúdo de texto atual do componente, possibilitando sua alteração.

Figura 5.28: Janela de Edição do Componente Texto de um OAC



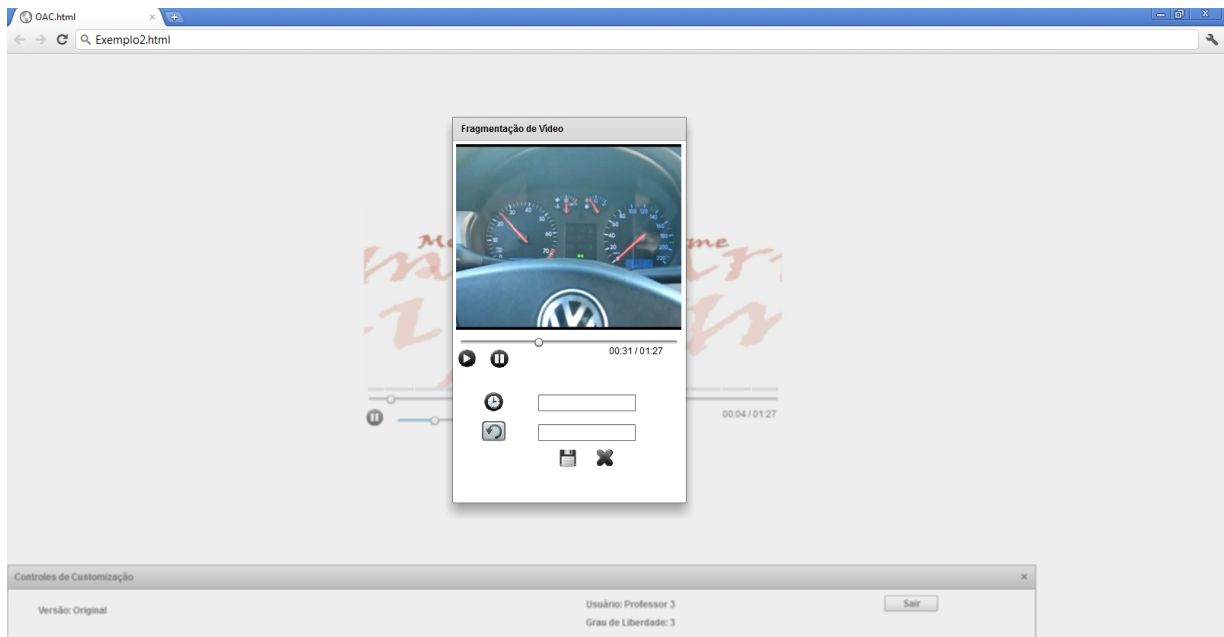
Fonte: Própria Autora

A fragmentação de vídeo é uma ação que permite delimitar os valores de apresentação de um elemento do tipo Vídeo. A Figura 5.29, apresenta um exemplo dessa janela.

### 5.5.2 Funcionalidades Adicionais

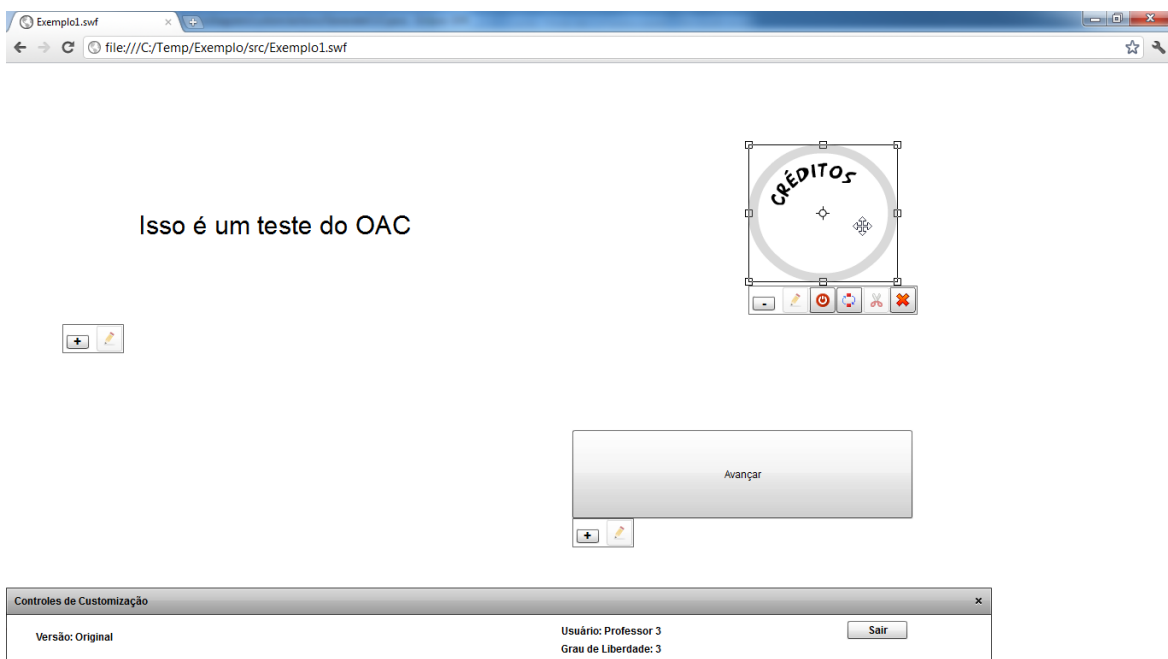
O usuário final de um OAC, gerado pelo *CLO Studio*, recebe um recurso com diversas funcionalidades adicionais para auxiliar o processo de customização. Por exemplo, ao solicitar uma ação de customização do tipo substituição de um elemento visual, o usuário também pode manipular o posicionamento e o dimensionamento do novo elemento. A Figura 5.30 mostra a execução dessa ação através da definição de um conjunto de âncoras ao redor do componente, que possibilita tanto reposicionamento como redimensionamento desse componente. Essa ação é realizada por uma ferramenta

Figura 5.29: Janela de Fragmentação de Vídeos de um OAC



Fonte: Própria Autora

de transformação da Senocular (vide Quadro 5.1) incorporada ao CLO *Studio*.

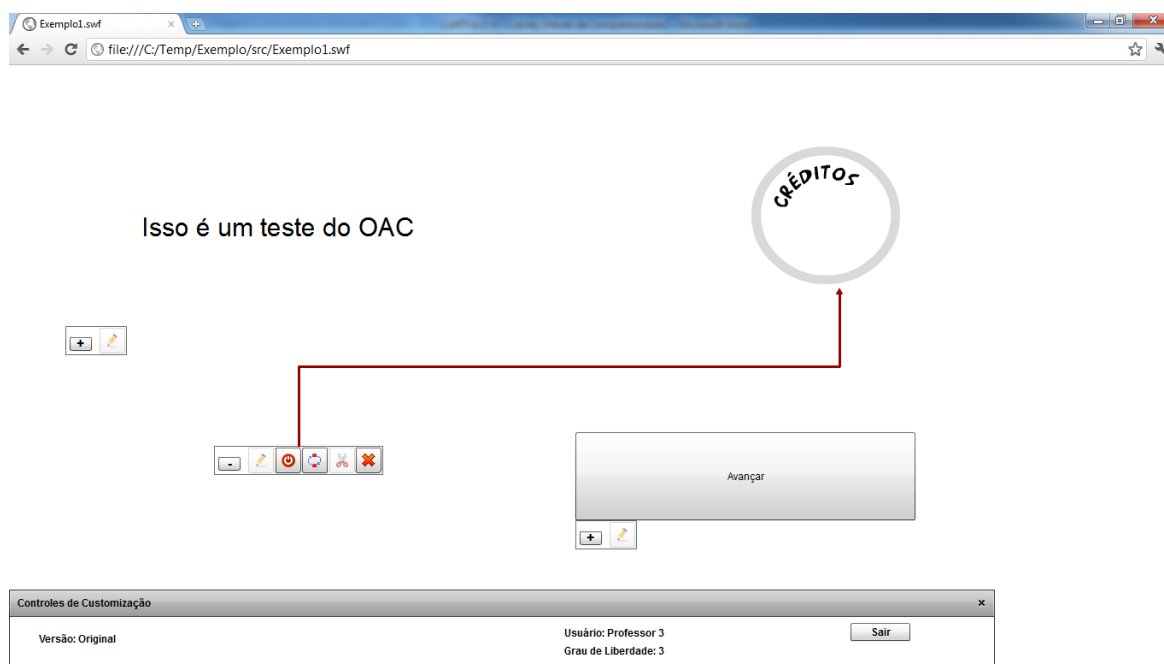
Figura 5.30: Ferramenta de Reposicionamento e Redimensionamento do CLO *Studio*

Fonte: Própria Autora

Além dessa ação de reposicionamento e redimensionamento, também foi incorporada ao *framework* de suporte do CLO *Studio* uma ferramenta de criação de conectores que possibilita a identificação de qual componente está associado a uma barra de con-

trole de customização. A Figura 5.31 mostra essa ferramenta em funcionamento. Como pode ser observado, uma seta (conector) aponta para o componente na qual as ações de customização serão realizadas. Essa ferramenta é de fundamental importância em cenas que possuem muitos componentes a serem customizados. Essa funcionalidade é oferecida pela implementação da classe `Connector.as`<sup>7</sup>.

Figura 5.31: Ferramenta de Conector de Componentes do CLO *Studio*



Fonte: Própria Autora

Todos os elementos adicionais utilizados na construção do *framework* de suporte do CLO *Studio* são transferidos diretamente no momento da geração do OAC. Assim, eles são incorporados à versão final compilada do recurso e estará disponível para o usuário no momento da execução do mesmo.

## 5.6 Discussão

O CLO *Studio* é uma DSL visual implementada como uma ferramenta de autoria de OAC. Isso significa que além de fornecer um conjunto de elementos visuais com os quais OAC podem ser descritos, essa ferramenta também oferece um conjunto de ações que implementam todo o processo CLAssROOM, apresentado no Capítulo 4. Portanto, essa

<sup>7</sup><http://dd.ourvfs.com/dd16ezra/WorldConstruction/misc/Connector.as>

ferramenta auxilia uma equipe de desenvolvimento de OAC durante todo o processo de desenvolvimento de um recurso, oferecendo suporte automatizado à diversas tarefas e propiciando uma maior cooperação entre os membros dessa equipe.

A notação utilizada no CLO *Studio*, para representar os elementos de um OAC, foi desenvolvida com a participação direta dos profissionais interessados. Essa notação congrega aspectos simples e precisos relacionados às diversas áreas de atuação desses profissionais, possibilitando a criação de projetos no qual todos os participantes podem incorporar suas sugestões e avaliá-las ainda em estágios iniciais do desenvolvimento.

Essa avaliação é ainda mais reforçada com a possibilidade de realização de simulações no CLO *Studio*. A operação de simulação apresenta uma versão “rascunho” do OAC e permite a interação dos profissionais como se fosse a versão final do recurso. Deste modo, as discussões relacionadas, por exemplo, ao posicionamento de elementos, ao sequenciamento de ações e à navegação entre cenas, podem ser simulados e eventuais problemas podem ser antecipados.

Essa operação de simulação leva essa DSL ao encontro do conceito de DSL executável (HEN-TOV; LORENZ; SCHACHTER, 2009; DJURIC et al., 2010). Esse tipo de DSL permite a “animação” das especificações com a intenção de possibilitar a visualização do funcionamento das aplicações antes dessas estarem finalizadas. Essa estratégia permite a antecipação da validação das aplicações (COMBEMALE et al., 2008), possibilitando uma análise mais detalhada de seus requisitos.

## 5.7 Conclusão

Neste capítulo foi apresentado o CLO *Studio*<sup>8</sup>, que é uma ferramenta de autoria, a ser utilizada por uma equipe de desenvolvimento de OA, no suporte à produção de OAC. Essa ferramenta introduz um conjunto de funcionalidades diferenciadas em relação a outras ferramentas de autoria de OA em geral, como, por exemplo, a simulação da execução de um OAC. Por ter sido produzida na plataforma Eclipse, o CLO *Studio* é de fácil utilização e possui uma linguagem bastante conhecida para os profissionais de

---

<sup>8</sup>Informações adicionais sobre o CLO Studio, assim como exemplos e vídeos de demonstração, podem ser encontrados em <https://sites.google.com/site/customizacaoguiada/>

---

projeto e desenvolvimento de aplicações. Adicionalmente, a notação utilizada para a criação dos projetos também torna essa ferramenta acessível a outros profissionais que participam do desenvolvimento desse tipo de recurso.

Os OAC, que são produtos finais gerados pelo *CLO Studio*, possuem uma interface de customização simplificada e cuja utilização é opcional. Desse modo, caso um professor não tenha conhecimento sobre o conceito de customização guiada para realizar as ações de customização, o recurso funciona como um OA monolítico. Assim, nenhuma interferência é introduzida no cotidiano desse professor. Contudo, se o professor possuir conhecimento sobre essa nova estratégia de adaptação, ele pode se beneficiar com a possibilidade de modificar os recursos de acordo com suas necessidades, observando as restrições impostas pela equipe de projeto no OAC e pelo grau de liberdade que ele possui.

## Capítulo 6

# Avaliação Empírica Qualitativa da Produção e Utilização de OAC

“Não é a intenção que valida um ato, mas seu resultado.”

– *Nicolau Maquiavel*

Neste capítulo são avaliadas as propostas apresentadas como contribuições desta tese. Nele são avaliadas, utilizando uma estratégia de Avaliação Empírica Qualitativa, tanto a fase de concepção e desenvolvimento quanto a fase de utilização dos OAC. Na fase de concepção e desenvolvimento são analisados o impacto do processo de produção de OAC para as equipes envolvidas nessa atividade, no qual são consideradas as interações estabelecidas entre os diversos profissionais que atuam nestas equipes. Na fase de utilização são analisados aspectos como autonomia dos professores no uso dos OAC.

Para apresentar essa avaliação o presente capítulo foi organizado em cinco seções. Na primeira, denominada Método de Avaliação, são apresentadas algumas informações sobre a avaliação empírica qualitativa assim como as justificativas da escolha dessa abordagem para este trabalho. Na segunda seção é apresentado o planejamento adotado para realizar as avaliações. Na terceira seção são apresentados a execução e a análise do dados coletados. Na quarta seção os resultados obtidos são discutidos e por fim, na quinta seção, é realizada uma conclusão.

## 6.1 Método de Avaliação

Os resultados produzidos por esta tese podem ser agrupados observando dois aspectos principais: concepção e desenvolvimento que representam a produção além da utilização de OAC.

Sob o ponto de vista da produção de OAC, o objetivo maior a ser analisado é o impacto do CLO *Studio* na comunicação entre os membros da equipe de desenvolvimento.

Observando o aspecto de utilização dos OAC é necessário avaliar o impacto desses recursos na autonomia dos docentes e na incorporação dos OAC nas suas práticas diárias.

Considerando que ambos os aspectos levantados tratam do papel humano frente a produtos de software, o método de avaliação adotado nesse trabalho é baseado na abordagem da Engenharia de Software Empírica (PERRY; PORTER; VOTTA, 2000; TICHY, 2000).

A abordagem empírica é utilizada em engenharia de software para avaliar se o que é observado é o que se crê como verdade (PERRY; PORTER; VOTTA, 2000). Essa área de pesquisa adota basicamente dois métodos para tratar as informações a serem avaliadas: o método quantitativo e o método qualitativo.

Segundo Easterbrook et al. (2008),

É importante compreender que em qualquer estudo empírico, as teorias têm um forte impacto sobre como as coisas são observadas e interpretadas. A teoria torna-se uma “lente” através da qual o mundo é observado. [...] Nos métodos de pesquisa *quantitativos*, a lente teórica é usada explicitamente para decidir quais as variáveis se devem isolar e medir, e quais ignorar ou excluir. Nos métodos *qualitativos*, a lente teórica é aplicada para categorização dos dados. (tradução da autora)

Conforme Seaman (1999), os métodos de pesquisa qualitativos foram desenvolvidos, principalmente, por pesquisadores da área de educação e outras ciências sociais, para



estudar a complexidade do comportamento humano (como por exemplo, motivação, comunicação e compreensão).

Dessa forma, para a avaliação dos resultados desse pesquisa foi utilizado o método de avaliação empírica qualitativa, visto que esse método se encaixa nas necessidades de avaliação dessa tese, pois os fatores mais importantes da avaliação dessa pesquisa tratam de aspectos comportamentais mediante os produtos de software resultantes deste trabalho.

### 6.1.1 Avaliação Empírica Qualitativa

Na última década a pesquisa empírica qualitativa vem se tornando um promissor e, ao mesmo tempo, confuso campo de estudo.(FLICK; KARDORFF; STEINKE, 2004) Esse método de pesquisa trata as informações sob o ponto de vista do comportamento das pessoas que fazem parte do processo, ao invés de medir o processo em si. Essa medição, aplicada nos métodos ditos quantitativos, tem sido amplamente utilizada em Engenharia de Software. Contudo, existem grandes diferenças nas duas abordagens, como explica APPLICATION (2004):

A [abordagem] quantitativa enfoca a relação entre a ciência social e a natural, valoriza a mensuração e o tratamento estatístico das variáveis como meios para a construção do conhecimento válido e generalizado. Na abordagem qualitativa, o enfoque, por sua vez, é na compreensão de um contexto particular, respaldando-se na interpretação, na busca de significado, na subjetividade e na intersubjetividade

Embora ambos os métodos tenham interesse de avaliar determinadas situações, a abordagem qualitativa não busca enumerar ou medir eventos e, geralmente, não emprega instrumental estatístico para análise dos dados; seu foco de interesse é amplo e parte de uma perspectiva diferenciada da adotada pelos métodos quantitativos.(NEVES, 1996)

Dessa forma, não é necessário a utilização de medidas (métricas) para realizar uma avaliação qualitativa, pois o foco principal está na análise dos fenômenos e não na

quantificação dos dados. visto que, segundo Kaplan e Maxwell (2005), a compreensão de um fenômeno do ponto de vista de seus participantes e seu particular contexto social e institucional é largamente perdido quando dados textuais são quantificados.

No entanto, mesmo com todo o avanço ocorrido nesse campo de pesquisa na última década, os métodos qualitativos ainda não são comumente utilizados em pesquisas em Engenharia de Software. Em vários estudos ditos “qualitativos”, os pesquisadores transformam em números os dados qualitativos coletados, contabilizando “números de respostas do tipo X” em comparação com “números de respostas do tipo Y”. Faz-se necessária uma maior discussão de como métodos de análise qualitativa podem ser utilizados em pesquisas em Engenharia de Software (CONTE; CABRAL; TRAVASSOS, 2009). Contudo, a força da pesquisa qualitativa é a sua capacidade para fornecer descrições textuais de como pessoas experimentam questões de pesquisa (hipóteses). (MACK et al., 2005)

Contudo, como os argumentos da pesquisa qualitativa são expressos sob a forma de texto, as diferenças de estilo ou de contexto, podem dificultar a captura da informação pelo pesquisador. (NEVES, 1996) Além disso, a tarefa de coletar e analisar os dados é extremamente trabalhosa e tradicionalmente individual.

Entretanto, a força e versatilidade desse método de pesquisa pode ser demonstrado pelos inúmeros trabalhos em diversas áreas do conhecimento, como por exemplo, educação (SAVENYE; ROBINSON, 1996; CRESWELL, 2002), saúde (POPE et al., 2005; TEIJLINGEN; FORREST, 2004), Ciências Sociais (STERK-ELIFSON, 1995; PHILIP, 1998) e Computação (MYERS; AVISON, 1997; WALSHAM, 1995).

Para a execução de uma avaliação empírica qualitativa Godoy (1995) aponta a existência de, pelo menos, três diferentes possibilidades oferecidas por essa abordagem: a pesquisa documental, o estudo de caso e a etnografia.

- A Pesquisa Documental é constituída pelo exame de materiais que ainda não receberam um tratamento analítico ou que podem ser reexaminados com vistas a uma interpretação nova ou complementar.
- Etnografia envolve longo período de estudo em que o pesquisador fixa residência

em uma comunidade e passa a usar técnicas de observação, contato direto e participação em atividades.

- O Estudo de Caso visa o exame de um ambiente, de um sujeito ou de uma situação em particular.

Essas possibilidades indicam a forma pela qual o pesquisador irá se inserir no processo de execução da pesquisa. Esse processo, por sua vez, pode ser dividido em duas grandes etapas (MYERS; AVISON, 1997): coleta e análise de dados.

#### 6.1.1.1 Métodos de Coleta de Dados

A coleta de dados é o primeiro passo a ser realizado na execução de uma pesquisa empírica qualitativa. Existem várias técnicas de coleta de dados para esse tipo de pesquisa disponíveis na literatura (TAYLOR; BOGDAN, 1998). Contudo, como apontado em Seaman (1999), podem ser destacadas as técnicas de Observação e de Entrevistas.

Na técnica de observação, ocorre a partir de interações sociais entre o pesquisador e o pesquisado nas quais os dados são coletados. A ideia principal dessa técnica é coletar a impressão do pesquisador sobre as ações dos pesquisados. Os dados capturados durante esse processo são registrados em “Diários de Campo”.

A realização de entrevistas também é uma técnica bastante utilizada. Essa técnica é aplicada com a intenção de registrar informações a partir da recordação das ações do entrevistado. As entrevistas podem ser dos seguintes tipos (WESTBROOK, 1994):

- Estruturadas: nesse tipo de entrevista as perguntas são pré-definidas e os resultados são apontados direta e explicitamente pelos entrevistados.
- Não-estruturadas: nessa abordagem tanto as perguntas como as respostas são originadas dos entrevistados. Neste caso as questões são completamente abertas e não há objetivo claro nos resultados.
- Semi-estruturadas: é uma mistura das abordagens anteriores. Nessa técnica tanto são coletadas informações pontuais e precisas como informações abertas

e não planejadas, fato que produz resultados mais relevantes quando utilizados em amostras de pequeno porte.

Para realizar a coleta de dados em uma avaliação empírica qualitativa, também é necessário a definição das amostras de usuários a serem utilizadas. Essa definição deve considerar as peculiaridades desse tipo de avaliação. Como apresentado em Onwuegbuzie e Leech (2007), a pesquisa qualitativa não necessita de análises estatísticas em relação à população, visto que o mais importante neste tipo de pesquisa é a obtenção de “*insights*” sobre práticas e processos em um determinado local e contexto. Ainda segundo os autores, a pesquisa qualitativa tipicamente envolve amostras de tamanho pequeno. Essa recomendação também é apresentada em Onwuegbuzie e Collins (2007), na qual os autores indicam que as amostras para a pesquisa qualitativa não podem ser tão grandes para não dificultar a extração de dados relevantes, nem podem ser tão pequenas que não consigam atingir a saturação dos dados <sup>1</sup>.

Uma técnica bastante aplicada na definição de amostras para pesquisa qualitativa é definida de grupos focais (KRUEGER; CASEY, 2009). Essa técnica, bastante utilizada em pesquisa de mercado qualitativa, se baseia na organização de grupos de indivíduos para a avaliação de determinadas informações. Uma vantagem dessa abordagem é a possibilidade de direcionar a pesquisa e extrair informações qualitativas importantes a partir das interações entre os indivíduos. Um inconveniente é o processo de recrutamento e composição dos grupos. Visto que esses devem seguir determinados critérios para serem formados.

Um primeiro passo para a composição desses grupos é a homogeneidade. Deste modo, critérios como ocupação, experiência no âmbito do problema, idade e sexo, devem ser considerados.

A definição dos tamanhos dos grupos também é um fator importante. Segundo Krueger e Casey (2009), tradicionalmente os grupos devem ter entre 10 e 12 pessoas. Contudo, esse autor informa que se os participantes possuem experiência no tópico pesquisado o grupo pode ter entre 5 e 7 participantes. Atualmente, com a demanda

---

<sup>1</sup>Saturação dos dados é o termo utilizado em pesquisa qualitativa para indicar que foi atingido um ponto na coleta dos dados na qual as informações estão se repetindo. Neste caso os dados não são mais importantes para a realização das análises. (MORSE, 1995)

de resultados rápidos de pesquisas qualitativas, tem sido utilizados os denominados mini grupos focais, definidos entre 4 e 5 participantes (IRELAND, 2003; KRUEGER; CASEY, 2009). Dentre os participantes dos grupos focais, deve sempre haver a figura de um moderador, que é o encarregado de organizar e coordenar as interações.

#### 6.1.1.2 Métodos de Análise de Dados

A análise dos dados é a etapa que se segue no processo de avaliação empírica qualitativa. Nesta etapa, os dados coletados passam por um tratamento para que sejam transformados em informação relevante para a pesquisa. Seaman (1999) agrupa os métodos de análise em dois tipos principais:

- **Método de Comparação Constante:** esse processo começa com a codificação das notas de campo e entrevistas. Essa codificação significa a identificação de informações, em trechos de texto ou áudio, que são relevantes para um determinado tema ou ideia de interesse do estudo. Em seguida, esses trechos são agrupados em tópicos de acordo com as informações que eles capturam. Esses agrupamentos são examinados para explicações dos fenômenos. O próximo passo é a elaboração de um memorando de campo que articula uma proposição (uma hipótese preliminar a ser considerado) ou uma observação sintetizados a partir dos dados codificados. O diário de campo destina-se a ser uma maneira informal de registrar rapidamente as descobertas do pesquisador antes que elas sejam perdidas.
- **Método de Análise Entre Casos:** neste método os dados são codificados em grupos diferentes e analisados de modo a procurar relações entre os mesmos. Essa estratégia é utilizada em análises que são feitas entre equipes diferentes de modo a confrontar as informações e verificar presença de inconsistências entre os dados.

#### 6.1.2 Método Adotado

Como apontado no início dessa Seção, a avaliação qualitativa empírica demonstrou ser a estratégia mais apropriada para avaliar as propostas desta tese. Para fazer uso dessa estratégia, nesta tese foram utilizadas duas etapas:

- **Planejamento da Avaliação:** nesta etapa foram definidos os materiais e instrumentos relacionados à avaliação empírica a serem utilizados e as estratégias de coleta de dados adotadas para cada avaliação, assim como algumas hipóteses norteadoras das análises.
- **Execução e Análise:** a aplicação das estratégias definidas no planejamento para a coleta de dados foram executadas nesta etapa. Nela também foram realizadas as análises dos dados coletados.

A realização dessa avaliação foi dividida em duas etapas. Na primeira foram avaliadas a Concepção e Desenvolvimento de OAC. Na segunda foi avaliada a Utilização dos OAC. É válido observar que tanto o Planejamento, descrito a seguir na Seção 6.2, como a Execução e Análise, descritos na Seção 6.3, apresentam as estratégias de avaliação para cada uma dessas etapas.

## 6.2 Planejamento

### 6.2.1 Materiais e Instrumentos Utilizados

Conforme detalhado na Seção 6.1, são apresentados no Quadro 6.1 os materiais e instrumentos para realizar as avaliações, tanto no que diz respeito à concepção e desenvolvimento quanto à utilização de OAC.

Quadro 6.1: Materiais e Instrumentos Utilizados na Avaliação Empírica Qualitativa

<b>Tipo de Pesquisa</b>	Estudo de Caso
<b>Técnica de Coleta de Dados</b>	Entrevista Semi-Estruturada
<b>Técnica de Análise de Dados</b>	Método de Comparação Constante
<b>Técnica de Amostragem</b>	Grupo Focal - Avaliação da Concepção e Desenvolvimento Mini Grupo Focal - Avaliação da Utilização
<b>Tamanho das Amostras</b>	Avaliação da Concepção e Desenvolvimento (6 pessoas) Avaliação da Utilização (4 pessoas)

Fonte: Própria Autora

A opção por um Estudo de Caso se deu pela necessidade de avaliar algumas situações em contextos específicos. Na avaliação da Concepção e Desenvolvimento foi

avaliada a utilização do CLO *Studio* na produção de um OAC pelas equipes pedagógica, *design* e programação. Na avaliação da Utilização foi avaliada a utilização de um OAC por um grupo de professores.

A técnica de coleta de dados utilizada foi a entrevista semi-estruturada. Para tanto, foram desenvolvidos dois roteiros de entrevista (Apêndices D.1 e D.2). O primeiro contém questões relacionadas à concepção e desenvolvimento de um OAC. O segundo contém questões que dizem respeito à utilização do recurso e são voltadas ao professor. Para a coleta de informações nas entrevistas foi utilizada a seguinte estratégia:

- Realização de anotações nos Diários de Campo, apresentados nos Apêndices E.1 e E.2. Esses diários foram criados com o objetivo de registrar as impressões da moderadora, como indicado na Seção 6.1.1, sobre a equipe de desenvolvimento, durante a fase de produção (concepção e desenvolvimento) do OAC, e sobre o professor, durante a fase de utilização do recurso customizável.
- Gravações das entrevistas: a gravação das entrevistas forneceu dados importantes para a compreensão de alguns momentos de interação entre os entrevistados.

A análise dos dados foi realizada pela técnica de comparações constantes. Para realizar essas comparações e suas respectivas anotações nos diários de campo, foram introduzidas rodadas de avaliação. Essas rodadas, denominadas neste texto como momentos de avaliação, permitiram um tratamento de dados de modo acumulativo. Desse modo, as avaliações realizadas em um primeiro momento, já haviam sido devidamente analisadas quando da execução de um segundo momento. Dessa forma, essas avaliações realimentavam o processo, fazendo com que as análises fornecessem informações mais relevantes.

Tanto a técnica de amostragem, quanto o tamanho das amostras, seguem a ideia de grupos focais. Essa técnica foi escolhida pelo fato dos avaliados fazerem parte de grupos homogêneos e com experiência anterior nas áreas de desenvolvimento e utilização de OA.

Nas Seções 6.2.3 e 6.2.4 são detalhados, respectivamente, os planejamentos da avaliação para a concepção e desenvolvimento e para a utilização de OAC.

## 6.2.2 Planejamento da Avaliação na Concepção e Desenvolvimento

Na etapa de concepção e desenvolvimento, a avaliação foi direcionada na construção dos OAC. Essa construção, como detalhado no processo CLAssROOM apresentado no Capítulo 4, ocorre através da criação dos modelos de arquitetura, projetados pela equipe pedagógica (pedagogos e especialistas de área), com eventuais intervenções realizadas pela equipe de *design* e de programação, que poderão sugerir ações que facilitem a utilização dos recursos por parte dos professores e alunos. Essas ações, no entanto, devem considerar apenas aspectos de navegação, de animações ou outros aspectos de natureza técnica, visto que a apresentação do conteúdo em si deve ser um trabalho, prioritariamente, desenvolvido pela equipe pedagógica.

A construção do modelo de arquitetura de um OAC é um trabalho realizado com o suporte do CLO *Studio*, que faz uso de uma DSL gráfica para nortear a construção desse modelo. Com o CLO Studio, à medida que o projeto é executado, os códigos dos OAC são gerados e, caso necessário, podem ser modificados pela equipe técnica.

Neste sentido, para avaliar essa etapa, deve ser observado o seguinte contexto:

1. O tempo de desenvolvimento<sup>2</sup> do OAC é desconsiderado, visto que a utilização de uma estratégia orientada a modelos no desenvolvimento deste tipo de RED possibilita a automatização de grande parte da geração de código para uma plataforma específica.
2. A concepção<sup>3</sup> do recurso é realizada através de interações entre os membros da equipe pedagógica, de *design* e programação. Essas interações permitirão a geração de um modelo de arquitetura do OAC que retrata tanto as características instrucionais (responsabilidade da equipe pedagógica) como elementos de interação e controle visual dos OAC.
3. O CLO *Studio* deverá ser compreendido por todos que participam do processo

---

<sup>2</sup>Entenda-se desenvolvimento como sendo a criação de códigos fontes em uma plataforma alvo pela equipe de desenvolvimento do OAC (equipe técnica).

<sup>3</sup>Entenda-se concepção como sendo a etapa na qual a equipe pedagógica, de design e técnica deve gerar um modelo de arquitetura através do CLO Studio.



de desenvolvimento do OAC, visto que essa ferramenta deve intermediar toda e qualquer interação entre os membros da equipe de produção desses recursos.

De modo a nortear a coleta e análise de dados dessa pesquisa qualitativa, foi definida uma motivação e um conjunto de hipóteses de pesquisa. Essas hipóteses devem ser avaliadas pelo moderador ao longo da execução da pesquisa.

### 6.2.2.1 Motivação e Hipóteses de Pesquisa

Como apresentado no Capítulo 4, nos processos de produção de um OA monolítico, as ações de interação são, normalmente, executadas através de troca de documentação específica para cada equipe que participa desse processo. Também deve ser considerado que esse potencial de interação é prejudicado com o avanço do desenvolvimento do recurso, visto que os artefatos gerados nas etapas mais adiantadas não são, na maioria das vezes, compreendidos por todos os membros dessas equipes. Assim, os códigos gerados pela equipe de desenvolvimento acabam servindo apenas para os desenvolvedores, não sendo facilmente mapeados em elementos passíveis de serem compreendidos e consequentemente avaliados pela equipe pedagógica, o que pode gerar “ruídos” ou falhas na interpretação dos requisitos e, deste modo, acarretar um custo adicional na realização de possíveis modificações nos recursos depois dos mesmos terem sido implementados.

No caso dos OAC, como apresentado no processo CLAssROOM proposto nesta tese, todo o processo de comunicação se dá através do modelo de arquitetura gerado pela DSL e implementado pelo CLO *Studio*. Por ser uma linguagem específica de domínio, a expressividade dessa linguagem minimiza a inserção de elementos que não sejam compreendidos pelas equipes de produção. Com isso, o CLO *Studio* representa uma forma de manter os interesses dos atores sincronizados durante todo o processo de desenvolvimento.

Sendo assim, o primeiro objetivo deste estudo foi avaliar se houve ganhos na comunicação ou interação dos membros das equipes pedagógica, de *design* e de programação envolvidos no processo de produção, intermediada pelo CLO *Studio*. Deste modo, a seguinte hipótese, definida na Seção 1.4 desta tese, deve ser avaliada:

---

**$H_A$ : A Engenharia de Software Orientada a Modelos possibilita uma melhoria na comunicação entre os membros da equipe de produção, favorecendo um melhor tratamento dos aspectos de customização guiada nestes recursos.**

---

Essa hipótese, por sua vez, foi desmembrada em hipóteses menores e, portanto, mais facilmente passíveis de comprovação. Desta forma, as seguintes sub-hipóteses podem ser levantadas:

$H_1$ : O modelo de arquitetura construído através da DSL é compreendido por todos os tipos de atores do processo.

$H_2$ : O CLO *Studio* é de fácil manipulação por todos os tipos de atores do processo.

### 6.2.2.2 Planejamento da Avaliação

A avaliação realizada deve considerar três etapas. A primeira está relacionada ao conhecimento do CLO *Studio*; a segunda está relacionada à manipulação dessa ferramenta e, para finalizar, na terceira são realizadas as entrevistas para a coleta e análise dos dados.

A etapa de conhecimento consiste na apresentação do CLO *Studio* para a equipe de desenvolvimento como ferramenta utilizada para auxiliar na concepção e criação de um OAC. Esta etapa tem por objetivo permitir que a equipe de desenvolvimento compreenda o funcionamento e as documentações geradas pela CLO *Studio*. Já a etapa de manipulação consiste em possibilitar que os membros da equipe de desenvolvimento construam um OAC com o uso do CLO *Studio*. Esta etapa tem por objetivo verificar se a utilização desta ferramenta na produção de um OAC acarreta alguma vantagem nas interações entre os membros da equipe.

Para realizar a avaliação, foi construído um grupo focal homogêneo seguindo às descritas na Seção 6.1.1.1. A composição desse grupo focal é descrita no Quadro 6.2.

Quadro 6.2: Estrutura do Grupo Focal de Avaliação da Concepção e Desenvolvimento

<b>Indivíduo</b>	<b>Ocupação</b>	<b>Experiência</b>	<b>Idade</b>	<b>Sexo</b>	<b>Representa.</b>
Moderadora	Coordenadora de Equipe de Desenvolvimento de OA	8 anos	36	F	MO
Especialista de Área	Professora de Matemática	6 anos	35	F	EA
Pedagoga	<i>Designer</i> Instrucional do Proativa	3 anos	23	F	PE
<i>Designer</i>	Designer Gráfico do Proativa	3 anos	20	M	DE
Programador	Desenvolvedor do Proativa	3 anos	20	M	PR

Fonte: Própria Autora

A composição desse grupo focal foi realizada procurando cobrir todos os membros de uma equipe de desenvolvimento do OA. A experiência indicada no Quadro 6.2 está relacionada ao conhecimento com OA. A Representação, apresentada na última coluna, foi utilizada na codificação<sup>4</sup> dos textos da pesquisa, indicando as fontes das informações levantadas.

### Conhecimento do CLO *Studio*

A etapa de conhecimento do CLO *Studio* foi realizada através de uma formação, ministrada pela moderadora, à equipe focal apresentada no Quadro 6.2.

Essa etapa, como apresentado na Seção 6.2.1, foi dividida, inicialmente, em 3 momentos:

**1º Momento:** Explicação do funcionamento e a notação utilizada no CLO *Studio* para projetar um OAC.

**2º Momento:** Explicação dos tipos de documentações geradas a partir do projeto do OAC e a quem elas se destinam. Neste momento, foram distribuídos exemplares das documentações entre todos os membros da equipe de projeto.

<sup>4</sup>Codificação é a ação de fornecer etiquetas para os dados levantados nas pesquisas qualitativas (KAPLAN; MAXWELL, 2005).

**3º Momento:** Explicação do funcionamento das ações de customização e como elas são inseridas no recurso durante a fase de produção.

Em seguida, a construção do OAC foi realizada na etapa de manipulação do CLO *Studio*.

### Manipulação do CLO *Studio*

Nessa etapa, também composta por três momentos distintos, a moderadora solicitou que os membros da equipe de desenvolvimento criassem um OAC com o auxílio do CLO *Studio*. Para isto, uma proposta de atividade foi fornecida. Essa proposta de atividade foi uma ideia inicial sobre o que desenvolver, de forma a direcionar as ações do grupo. Essa proposta foi descrita através de um roteiro de atividades incompleto, de modo que os especialistas de área e pedagogos pudessem preencher as lacunas com suas ideias. Uma vez a proposta totalmente preenchida, a mesma foi projetada usando o CLO *Studio*.

Posteriormente, o *designer* utilizou o roteiro (*storyboard*) gerado pelo CLO *Studio* para construir os componentes que formavam as interfaces do OAC. Esses componentes uma vez projetados foram disponibilizados para a geração de códigos do OAC pelo CLO *Studio*. Alguns ajustes também foram solicitados para a equipe técnica nos arquivos MXML gerados por essa ferramenta.

O projeto das interfaces do OAC, bem como os ajustes realizados no mesmo, foram feitos com o suporte do roteiro (*storyboard*) e através de interações estabelecidas entre a pedagoga, o *designer* e o programador. Essas interações serviram para avaliar o processo de comunicação entre esses profissionais.

Encerrada a etapa de manipulação do CLO, a moderadora coletou os dados de pesquisa através das entrevistas.

### Entrevistas

Após a criação do OAC, foram aplicadas questões contidas no Roteiro de Entrevista da Equipe de Desenvolvimento, descritas no Apêndice D.1, com cada um dos repre-

sentantes das equipes (pedagógica, *design* e de programação) envolvidas na formação e construção do recurso.

Na Seção 6.2.3 é apresentado o planejamento da avaliação realizado na fase de utilização.

### 6.2.3 Planejamento da Avaliação na Utilização

O conceito de customização guiada, proposto e apresentado no Capítulo 3 desta tese, tem sua maior expressividade na manipulação do recurso pelo professor. De fato, a possibilidade de realizar modificações em um OA sem a obrigatoriedade de manipular seu código fonte, abre novas possibilidades na exploração deste tipo de recurso. Assim, o objetivo principal dessa avaliação é analisar a autonomia do professor para realizar essas modificações em um OAC.

Para que essa avaliação seja realizada, deve ser observado o seguinte contexto:

1. Os professores devem ter conhecimento em utilizar OA monolíticos (convencionais) em suas estratégias pedagógicas. Desse modo, os professores que participam dessa avaliação devem ter utilizado OA não customizáveis em suas aulas e, portanto, devem compreender como esses recursos podem dar suporte à exploração dos conceitos que os mesmos abordam.
2. Os professores entrevistados não devem ter participado do processo de desenvolvimento dos OAC, para que não tenham uma visão prévia das ações de customização a serem realizadas neste tipo de recurso. Ações essas que foram especificadas por especialistas, tanto na fase de concepção, quanto na etapa de definição do conteúdo abordado. Isto implica dizer, que os professores participantes da avaliação devem apenas manipular os recursos e não ter nenhuma ideia pré-concebida sobre as possibilidades de ações de customização existentes nos mesmos.

Observando o contexto no qual um OAC deve ser utilizado, a avaliação da postura docente diante deste recurso foi realizada com base na motivação descrita a seguir.

### 6.2.3.1 Motivação e Hipóteses de Pesquisa

Considerando o exposto anteriormente, a avaliação dos OAC na fase de utilização, teve como motivação a análise da postura dos professores em relação às novas possibilidades que esses recursos oferecem. Isso implica avaliar, especificamente, a efetividade das ações permitidas por esses recursos no desenvolvimento da autonomia do professor. Assim, a aferição dessa autonomia deve ser a principal meta a ser atingida na utilização de um OAC.

Deste modo, a hipótese descrita abaixo, definida na Seção 1.3 desta tese, deve ser avaliada.

---

***H<sub>B</sub>: A utilização de recursos adaptáveis (OA customizáveis) em sala aula eleva a autonomia dos professores na criação de novas oportunidades e estratégias de ensino***

---

A ideia de autonomia apresentada em  $H_B$  deve versar sobre a possibilidade de o professor realizar as ações por si mesmo, sem a intervenção de terceiros. Além disso, o professor deve vislumbrar novas oportunidades de utilizar o mesmo recurso em outros momentos através da possibilidade de criar versões customizadas dos mesmos.

Assim como na fase de concepção, a hipótese apresentada nesta fase pode ser desmembrada em hipóteses menores e, portanto, mais facilmente passíveis de comprovação. Desta forma, as seguintes sub-hipóteses devem ser avaliadas:

$H_3$ : Os elementos disponíveis para acessar as ações de customização na interface de um OAC são intuitivos e de fácil manipulação.

$H_4$ : As ações de customização e suas respectivas consequências podem ser facilmente compreendidas.

$H_5$ : Novas versões dos recursos podem ser criadas a partir da customização de um OAC para responder a novas necessidades específicas.

Para avaliar as hipóteses levantadas na utilização do OAC, foi desenvolvido um planejamento para nortear essa avaliação.

### 6.2.3.2 Planejamento da Avaliação

A avaliação nessa fase teve por objetivo mostrar ao professor de que modo ele poderá adaptar os elementos que compõem um OAC e, além disso, fazê-lo entender de que forma essas adaptações serão úteis para sua prática docente. Isso foi feito através da manipulação de um OAC.

Para a realização dessa avaliação foi construído um mini grupo focal formado com a estrutura descrita no Quadro 6.3. A experiência descrita no Quadro 6.3 está relacionada a utilização de RED no contexto de sala de aulas.

A avaliação da utilização foi organizada em três etapas distintas. Na primeira foi realizada uma entrevista de sondagem com os professores para avaliar as experiências desses profissionais na utilização de OA monolíticos em suas práticas. Na segunda etapa, além da apresentação de um OAC e das ações de customização que esses podem receber, foi apresentado um OAC para que os professores pudessem realizar intervenções seguindo algumas situações pré-definidas. Na terceira etapa foi realizada uma nova rodada de entrevista para coletar dados sobre a utilização dos OAC nos contextos apresentados. Essas etapas são detalhadas a seguir.

#### Entrevista de Sondagem

Foram aplicadas as questões contidas no Roteiro de Entrevista do Professor, descritas no Apêndice D.2.

#### Manipulação dos OAC

Após a entrevista realizada na primeira etapa, é importante mostrar ao professor, de modo indireto, algumas ações de customização (desabilitar, habilitar, esconder, mostrar editar, substituir) passíveis de serem realizadas com os elementos que compõem um OAC. Para isto, devem ser apresentadas ao docente, situações que requerem

Quadro 6.3: Estrutura do Mini Grupo Focal de Avaliação da Utilização

<b>Indivíduo</b>	<b>Ocupação</b>	<b>Experiência</b>	<b>Idade</b>	<b>Sexo</b>	<b>Representa.</b>
Moderadora	Coordenadora de Equipe de Desenvolvimento de OA	8 anos	36	F	MO
Professora de Ensino Superior	Doutoranda em Educação da Pontifícia Universidade Católica (PUC/SP) e mestre em Tecnologia da Informação e Comunicação na Formação em EaD pela Universidade Norte do Paraná em parceria com a Universidade Federal do Ceará	6 anos	35	F	PA
Professora de Ensino Superior	Mestrado e doutorado em Educação pela Universidade Federal do Ceará	5 anos	30	F	PB
Professora de Educação de Jovens e Adultos (EJA)	Licenciatura em Letras	2 anos	40	F	PC

Fonte: Própria Autora

adaptações em um OAC, para serem utilizadas no contexto apresentado. Nesta tese, foi utilizado o cenário FAZENDA do OAC Histórias Fantásticas<sup>5</sup>. Este, por sua vez, foi originado a partir do OA Histórias Fantásticas (HF), desenvolvido pelo grupo de pesquisa e produção de ambientes interativos e objetos de aprendizagem Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (PROATIVA) da Universidade Federal do Ceará.

Nesse momento, foi dado ao professor um contexto no qual era necessário a realização de adaptações no cenário do OAC. Também foram apresentados um conjunto de

<sup>5</sup> <http://www.proativa.virtual.ufc.br/oa/historias/>



situações a serem trabalhadas com base neste contexto. O Quadro 6.4 apresenta esse contexto e as situações apresentadas.

Quadro 6.4: Contexto e Situações para a Avaliação na Utilização

<b>Contexto</b>	<b>Situação</b>
Professor (a), suponha que você necessite trabalhar a produção textual com seus alunos. O gênero textual escolhido será uma fábula e para isso você contará com a ajuda do OAC Histórias Fantásticas. Com base nessas informações, analise cada uma das situações fornecidas e indique a(s) ação(ões) necessária(s) para adaptar o OAC a esse contexto.	No OA Histórias Fantásticas você dispõe de seis cenários. Qual deles não deve ser apresentado aos alunos para criação da fábula, uma vez que seu contexto não permite a produção deste tipo de gênero textual?
	Utilize o cenário FAZENDA para explicar o gênero fábula aos alunos. Que ação deve ser executada com os outros quatro cenários para que você consiga centrar a atenção dos alunos somente no cenário indicado?
	No cenário FAZENDA, você dispõe de botões contendo personagens, objetos e animais. Se você irá propor a criação de uma fábula a seus alunos, que mudanças nos botões necessitam ser feitas?
	Com a retirada de um dos botões do cenário fazenda, que tipo de ajuste será necessário fazer no botão AJUDA?

Fonte: Própria Autora

A terceira etapa foi realizada ao final da execução de todas as situações pelos professores. Nesta etapa, uma segunda rodada de entrevista foi realizada, onde os professores descreveram suas impressões sobre as ações de customização e os tipos de contribuições que os OAC puderam fornecer às suas atividades.

## Entrevistas

Nesta etapa foram aplicadas as questões contidas no Roteiro de Entrevista do Professor, descritas no Apêndice D.2.

## 6.3 Execução e Análise

Nesta seção é apresentado de que modo os planejamentos definidos na Seção 2.5 foram executados e a forma como os resultados dessas execuções foram analisados a partir dos dados coletados.

### 6.3.1 Execução da Avaliação na Concepção e Desenvolvimento

A execução da avaliação na concepção e desenvolvimento, como planejado, foi executada em seis momentos distintos, sendo que cada um desses teve a duração de três a quatro horas e periodicidade uma vez por semana durante os meses de novembro e dezembro de 2011

#### Explorando cada momento

Os três primeiros momentos são referentes à etapa de conhecimento da ferramenta CLO *Studio* pela equipe de desenvolvimento, enquanto os demais descrevem o processo de manipulação da referida ferramenta.

**1º momento:** Consistiu na apresentação tanto da ferramenta CLO *Studio* quanto da sintaxe utilizada na DSL. Neste momento, a moderadora criou um projeto simples utilizando o CLO *Studio* para que os membros da equipe de produção observassem na prática como utilizar a DSL e, com isso, compreender qual era de fato, o objetivo e a proposta da ferramenta.

**2º momento:** Utilizando o projeto criado no 1º momento, foi apresentada à equipe de produção as ações de customização e de que forma, com o uso do CLO *Studio*, era possível atribuir as permissões a cada um dos elementos do projeto. A preocupação da moderadora era fazer com que a equipe não apenas compreendesse como tornar um OA customizável, mas também compreendesse a importância e as implicações que essas intervenções podiam acarretar na prática docente.

**3º momento:** Para que os membros da equipe de produção compreendessem de fato como manipular o CLO *Studio*, a moderadora entregou uma proposta inicial de um OAC a ser projetado e, conseqüentemente, desenvolvido pela equipe com o

auxílio da ferramenta. A escolha por entregar uma proposta inicial à equipe foi para direcionar melhor as atividades, principalmente, da especialista de área e pedagoga. Nesta proposta foram informados a área e o conteúdo a ser trabalhado, deixando de livre escolha, o nível a ser empregado e a forma como o conteúdo deveria ser abordado.

Neste momento, a equipe concentrou esforços para delimitar o escopo do OAC bem como em manusear o *CLO Studio*.

**4º momento:** A equipe continuou projetando e realizando ajustes no OAC com o auxílio do *CLO Studio* e inseriu as permissões de customização nos elementos.

**5º momento:** A equipe realizou os últimos ajustes no projeto, com base na simulação do *CLO Studio* e, em seguida, enviou o roteiro gerado pela ferramenta para o *designer* e programador providenciarem, respectivamente, as imagens dos cenários e os ajustes necessários no código fonte.

**6º momento:** De posse das imagens e ajustes no código, a equipe visualizou o OAC gerado.

Após o 6º momento, a moderadora fez uso de um roteiro de entrevista semi-estruturada no intuito de capturar as impressões de cada um dos membros sobre a produção do OAC a partir do modelo de arquitetura gerado pelo *CLO Studio*. Neste roteiro foram realizadas perguntas direcionadas especificamente a cada um dos membros da equipe de produção, como por exemplo:

- A linguagem adotada na DSL e utilizada no *CLO Studio* foi de fácil compreensão?
- A notação empregada na DSL atendeu suas necessidades para projetar o OAC?
- O *CLO Studio* possibilitou criar uma clara representação da estrutura do OAC?
- É fácil manipular os elementos da DSL no *CLO Studio*?
- As ações de customização foram facilmente compreendidas e utilizadas?

- A documentação gerada pelo CLO *Studio* possibilita compreender de modo claro o sequenciamento das ações de navegação contidas na estrutura do OAC?
- A comunicação entre vocês, sofreu alguma interferência (não alterou/melhorou/-piorou) com o uso da documentação gerada pelo CLO *Studio*?

Por esta avaliação ter sido baseada em um Estudo de Caso, como apontado no Quadro 6.1, os momentos descritos anteriormente foram organizados com a intenção de construir um OAC de modo a criar um contexto no qual as respostas para os questionários de avaliação pudessem ser avaliadas.

### 6.3.1.1 OAC Produzido

Como mencionado anteriormente, o Estudo de Caso aplicado nessa avaliação produziu um OAC denominado Pizza\_Fração.

Como o próprio nome diz, o OAC Pizza\_Fração tem por objetivo fazer uso de uma pizza para trabalhar o conceito de fração através do significado parte-todo<sup>6</sup>.

Desenvolvido para ser trabalhado em séries iniciais do ensino fundamental (5<sup>o</sup> e 6<sup>o</sup> ano), o OAC busca mostrar a influência do numerador e denominador de uma fração, dentro de um contexto.

O contexto trabalhado no referido OAC consiste da produção de pizza por um *pizzaiolo* que pede ajuda ao usuário para entregar os pedidos aos clientes. As informações sobre os pedidos serão repassadas ao usuário. Este, por sua vez, indicava em quantas fatias a pizza deveria ser fatiada e quantas dessas deveriam ser entregues ao cliente. Para ajudar o *pizzaiolo*, o usuário tem disponível no OAC, o botão fatiar pizza. Este botão tem a função de dividir a pizza, conforme o pedido apresentado ao usuário. Essa ação de dividir, indica o denominador da fração. Para que o usuário selecione a quantidade de fatias desejada pelo cliente e, com isso, consiga entregar o pedido corretamente, o OAC dispõe da ação de selecionar fatias, sinalizando quantas fatias devem ser entregue ao cliente. A ação de selecionar, indica o numerador da fração. É importante ressaltar que caso o usuário faça uso da ação de selecionar antes de fatiar

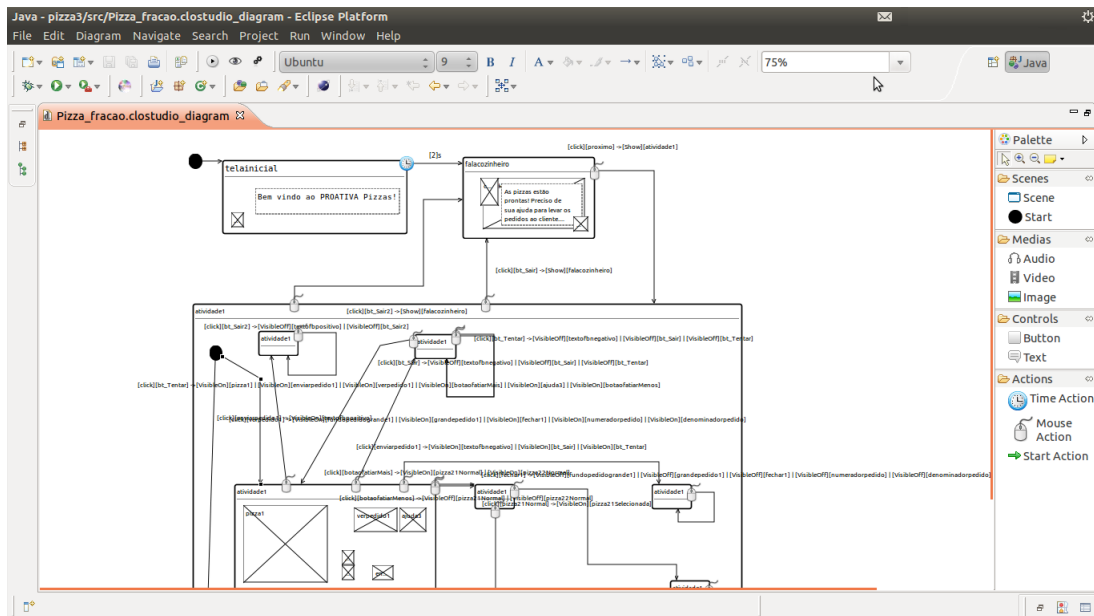
---

<sup>6</sup> Partição de um todo em n partes iguais, em que cada parte pode ser representada como  $1/n$

a pizza, será informado, via *pop up*, que a ação de fatiar só poderá ser realizada após a pizza ser fatiada.

Para a criação do projeto, a equipe pedagógica (especialista de área e pedagoga) fez uso do *CLO Studio*. Conforme apresentado na Figura 6.1. O projeto desenvolvido contém uma única atividade, porém muitas subcenas.

Figura 6.1: Projeto do OAC Pizza\_Fração no CLO Studio



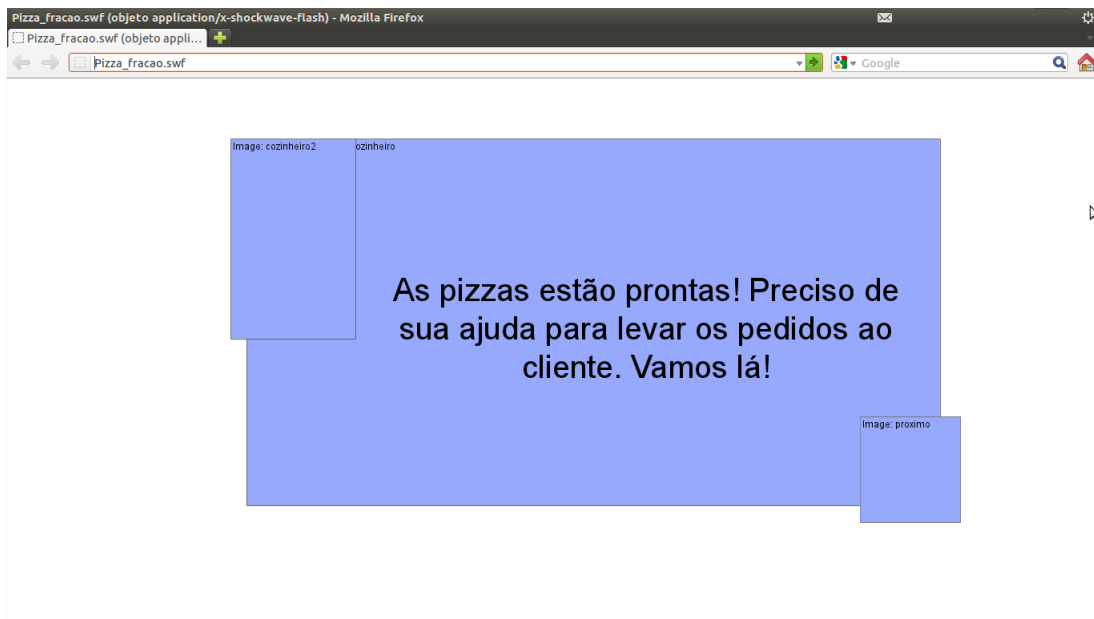
Fonte: Própria Autora

Ainda com o auxílio do *CLO Studio*, à medida que a equipe pedagógica projetava o OAC Pizza\_Fração, o mesmo era simulado como forma de identificar alguma inconsistência dentro do projeto.

A Figura 6.2 apresenta a simulação da cena “falacozinheiro” que é a primeira cena após a cena “telainicial”. Nela é possível observar que todos os elementos que foram projetados na referida cena estão presentes na simulação em formato de caixas como, por exemplo, a imagem do cozinheiro, o texto *pop-up*, o fundo do *pop-up*, além da imagem do botão “próximo”.

Finalizado o projeto foram realizadas as simulações necessárias para garantir que o mesmo correspondesse exatamente ao que foi especificado. Posteriormente o roteiro gerado, apresentado no Apêndice F, foi repassado ao *designer* e ao programador, juntamente com os arquivos em formato MXML, para que as imagens fossem criadas e os

Figura 6.2: Simulação do OAC Pizza\_Fração no CLO Studio



Fonte: Própria Autora

ajustes necessários no código fossem feitos.

De posse das imagens e dos ajustes realizados, o OAC ficou pronto para ser utilizado, conforme apresentado na Figura 6.3.

Figura 6.3: OAC Pizza\_Fração em Execução



Fonte: Própria Autora

### 6.3.1.2 Resultados Alcançados

Como indicado nos materiais utilizados no processo de avaliação (Seção 6.2.1), foram realizadas gravações nas quais foram registradas as impressões dos avaliados. Adicionadas às informações registradas nos Diários de Campo pela moderadora, esses dados foram organizados em tópicos, como sugere a técnica de codificação, utilizada no método de análise de comparações constantes. A seguir são apresentados os tópicos utilizados, assim como as transcrições dos trechos das gravações e as impressões da moderadora sobre os mesmo.

#### **Tópico:** Abertura para os Professores

Um aspecto importante da customização guiada diz respeito à abertura fornecida aos professores para modificar um recurso. Realizar modificações nos OA atualmente requer a manipulação direta dos códigos fontes desses recursos. A abordagem do CLO *Studio* possibilita a introdução de ações de customização nos elementos dos OAC de forma que os professores possam ter opções de modificação sem, no entanto, ter que manipular diretamente os códigos do recurso. Sobre essa abertura foram coletadas as seguintes informações:

[PR] “é muito difícil encontrar um professor que consiga mexer nos fontes do OA, pra não dizer quase nenhum.”

[EA] “imagine o professor mexendo no código do recurso (risos).”

[PR] “...o professor só precisa abrir o recurso e ter criatividade para modificá-lo.”

**Impressões da Moderadora:** a disponibilização dos códigos fontes dos OA não se enquadrava na abertura ideal para os professores. Com os OAC, os professores não terão mais as limitações proporcionadas pelos OA monolíticos, visto que as modificações necessárias são proporcionadas pelas ações de customização que foram sinalizadas, ainda na fase de projeto, pela equipe pedagógica.

#### **Tópico:** Representação de Informações pela DSL

O CLO *Studio* é a implementação de uma ferramenta visual de uma linguagem específica de domínio (DSL). Assim, essa DSL possui uma notação específica para representar os elementos presentes em um OAC. Sobre essa representação de informações do CLO *Studio* foram levantadas as seguintes informações:

[PE] “tranquilamente se constrói um OA com a notação usada na DSL. Rapidamente tiramos você (moderadora) de cena . Tá bem claro.”

[PR] “[O CLO *Studio*] ... faz algo que a gente não conseguia fazer que era a representação daquilo que estávamos pensando.”

[EA] “o esconder é ótimo porque ai eles (alunos) não vão usar outra atividade que não tenha sido a planejada.”

[EA] “Simples demais. Tranquilamente se constrói um OA com a notação usada na ferramenta. Rapidamente tiramos você [moderadora] de cena e ficamos eu [EA] e [PE] brigando pelo mouse (risos). Tá bem claro.”

**Impressões da Moderadora:** a notação empregada na DSL atende às necessidades para projetar um OAC, visto que ela representa claramente a estrutura desse recurso.

**Tópico:** Manipulação do CLO *Studio*

Durante o 1º momento, foi ressaltado, junto ao grupo avaliado, que o CLO *Studio* é uma ferramenta de autoria que tem o propósito de facilitar a construção de OAC. Apesar da maioria das ferramentas de autoria serem desenvolvidas especificamente para o professor ou qualquer outro usuário que tenha interesse em desenvolver um recurso, o CLO *Studio* é uma ferramenta que foi desenvolvida para ser utilizada por uma equipe de produção de OAC, considerando aspectos de multidisciplinaridade dessa equipe. Sobre a manipulação do CLO *Studio* pelos diferentes profissionais dessas equipes foram levantadas as seguintes informações:

[EA] “É realmente fácil mexer na ferramenta.”



[PR] “É como se você conseguisse mexer com a mão no recurso. Está todo mundo falando o mesmo algoritmo.”

[PR] “As modificações com o CLO é bem mais rápido e mais concreto.”

[PE] “as subcenar parecem um negócio estranho. Estranho não porque é difícil fazer ou tá errado, mas porque é difícil entender. Principalmente porque a visualização não é muito boa. O que favorece a compreensão são as conexões e é partir delas que se consegue entender.”

**Impressões da Moderadora:** o grupo avaliado ressaltou que, após a apresentação do CLO *Studio* pela moderadora, a utilização da ferramenta foi intuitiva e que sua utilização não exigiu nenhum conhecimento extra por parte dos membros do grupo. Muito embora a criação de subcenar tenha envolvido um grau maior de complexidade no que se refere à compreensão desse conceito, todos foram unânimes em dizer que com a prática essa estrutura será melhor compreendida.

**Tópico:** Criação de um novo OAC no CLO *Studio*

O CLO *Studio* é uma ferramenta que oferece suporte à produção de OAC desde a sua concepção até sua implementação. Sobre a utilização dessa ferramenta no suporte a esse processo foram levantadas as seguintes informações:

[PR] “... principalmente depois que você produz você gera a simulação. Você ter a possibilidade de ver a navegabilidade antes do OA ser desenvolvido, é maravilhoso. Com a simulação você já sabe o que vem em seguida.”

[EA] “... as “caixinhas” da simulação auxiliam muito a projetar corretamente o recurso. Já para o programador, as “caixinhas” facilitaram não só a imaginação, mas também toda a visualização do fluxo do recurso.”

[EA] “eu não sei mensurar o tempo, mas eu não tenho nem dúvida que irá diminuir e muito o tempo [de desenvolvimento]”

[EA] “A estória do retrabalho vai diminuir, porque com o CLO a gente já sabe o que tá certo e o que tá errado.”

[PE] “com o CLO *Studio* eu posso estruturar onde eu quero cada elemento”

[PE] “Com a visualização da figura fica mais fácil detectar um possível erro”

[DE] “aumenta a produtividade. Com o CLO *Studio* é preciso. Já é fechadinho”

[PR] “fácil imaginar todo o fluxo do OA só a partir das caixinhas.”

**Impressões da Moderadora:** a capacidade de simular um projeto, ainda em fase de produção, foi um ponto relevante da avaliação. Essa funcionalidade do CLO *Studio* dá condições de realizar ajustes antes de o recurso ser desenvolvido. O CLO *Studio* afeta positivamente a produtividade das equipes, visto que o tempo de desenvolvimento pode ser diminuído e os erros podem ser detectados ainda na concepção do produto.

**Tópico:** Participação das Equipes

A produção de um OAC é uma tarefa que exige a participação de equipes formadas por profissionais de diferentes formações e com interesses específicos e complementares. Essas equipes devem trabalhar de forma harmônica para que o recurso satisfaça a todos os interesses envolvidos. Sobre o aspecto de participação das equipes na produção de um OAC foram levantadas as seguintes informações:

[PE] “eu vou estar em ação, na prática, porque eu visualizo minha cena. No texto eu vou apenas supor e que na verdade não se sabe se essa suposição dará certo [referenciando quando se utilizava somente o roteiro feito em linguagem natural para produzir um OA]. Com o CLO, a ideia do que você está pensando funciona.”

[DE] “...as decisões são mais concretas, porque todo mundo está visualizando a mesma coisa e falando o mesmo algoritmo.”

[DE] “... o projeto com o CLO incentiva o trabalho colaborativo.”

[EA] “... o roteiro gerava muito retrabalho. Quando chegava [roteiro] na equipe de *design*, eles achavam que tinham entendido, mas quando retornava pra gente nem sempre correspondia ao que nós tínhamos planejado”.

**Impressões da Moderadora:** a estratégia de construção coletiva do CLO *Studio* facilita a especificação de ações que precisam ser compreendidas pelos diferentes profissionais das equipes. O documento de roteiro gerado por essa ferramenta passou a ser um ponto de convergência dos esforços das equipes, sendo que quaisquer discordâncias podem ser resolvidas ainda em estágios iniciais da produção dos recursos.

### 6.3.2 Execução da Avaliação na Utilização

Considerando o planejamento descrito na Seção 6.2.3, a execução da avaliação na utilização foi dividida em três etapas. Na primeira, foi realizada uma entrevista de sondagem para conhecer a experiência dos avaliados sobre a utilização de OA em sala de aula. Na segunda etapa foi apresentado um OAC aos avaliados e foram introduzidos um contexto de situações que estimulassem a necessidade de realizar adaptações neste recurso. Em seguida, na terceira etapa, foi realizada uma entrevista com o grupo para colher as informações sobre a utilização desse recurso.

#### 6.3.2.1 OAC Utilizado

Para realizar a avaliação das ações de customização por parte dos docentes, foi utilizado o OAC Histórias Fantásticas. Este recurso foi produzido, como mencionado na subseção 6.2.3, a partir do OA Histórias Fantásticas (HF), produzido pelo PROATIVA<sup>7</sup>. Desenvolvido para ser utilizado nas séries iniciais do ensino fundamental, o OA HF possui duas atividades (Atividade 1 e Atividade 2), ambas com o objetivo de servir de ferramenta auxiliar na produção textual. A primeira atividade, fornece um conjunto de cenários (FAZENDA, PRAIA, CIRCO, FLORESTA, SERTÃO, ESCOLA) para estimular a criação de textos. Já a segunda, deixa que o usuário exercite sua imaginação

<sup>7</sup> <http://proativa.virtual.ufc.br/oa/historias/>

não só na produção do texto, mas também na criação de imagens e cenários que representem o texto construído. A avaliação foi realizada utilizando o cenário FAZENDA da Atividade 1.

É importante ressaltar, que devido a complexidade do OAC produzido na avaliação da concepção e desenvolvimento, este não foi finalizado em tempo para ser utilizado na avaliação da utilização. Motivo pelo qual, optou-se por utilizar o OAC HF.

### **Explorando cada etapa**

**1ª etapa:** Consistiu na sondagem do conhecimento dos professores a cerca da utilização do OA monolítico, contemplando aspectos que justificam adoção do referido recurso em sala de aula. Para dar suporte aos esclarecimentos fornecidos, foram inseridos no roteiro de entrevistas perguntas do tipo:

- Dê exemplo de um OA que utilizou em sala de aula
- Por que você utilizou o OA em sua aula?
- Como você inseriu o OA em seu planejamento de aula?
- Que dificuldades você sentiu na aplicação do OA em sala de aula (dificuldades pessoais e dos alunos)?
- Você já deixou de utilizar um OA em sala de aula porque algum aspecto (linguagem, representação, níveis de atividades etc.) dele não atendia seu público? Qual?

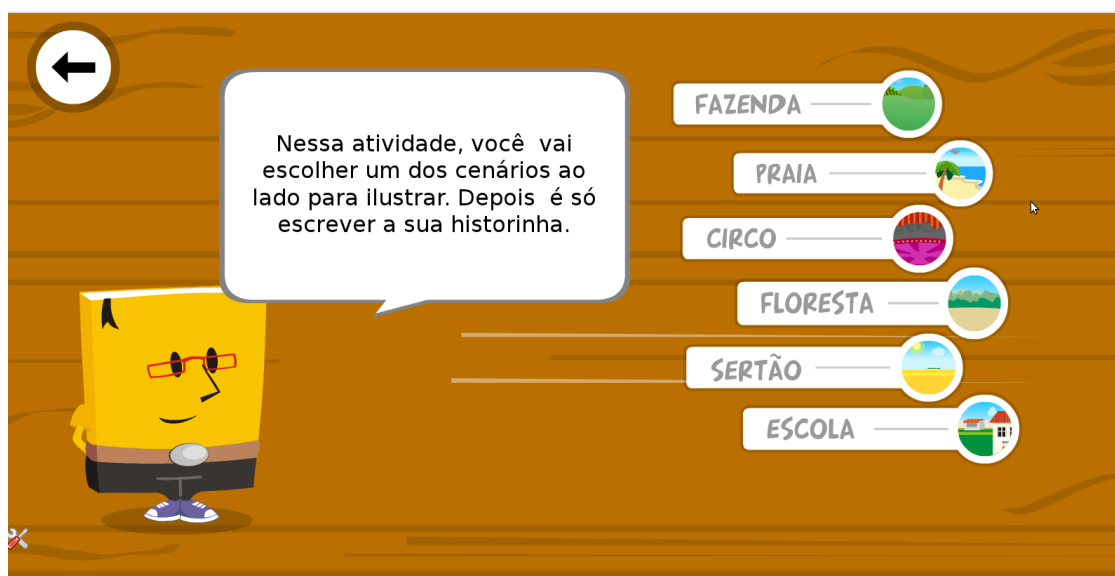
**2ª etapa:** O 1º momento dessa etapa deu início com a apresentação do OAC HF, pela moderadora, a cada uma das docentes. Essa apresentação teve por finalidade mostrar às docentes a categoria dos OA customizáveis e o modo como as alterações na interface do recurso, a partir dos ícones de customização, podiam ser realizadas, além de fazê-las perceber os reais benefícios de se trabalhar com recursos adaptáveis em sala de aula.

Essa percepção ocorreu a partir da exploração do OAC pelas docentes, em que cada uma delas realizou várias alterações na interface do recurso. Dentre as alterações realizadas, podem ser citadas desabilitar, esconder, substituir e editar elementos. Não foram realizadas alterações que necessitassem fragmentar vídeo, devido à ausência desse elemento no OAC HF. Concluída essa etapa, foi iniciado o 2º momento da avaliação.

Esse 2º momento consistiu na apresentação de um contexto planejado pela moderadora. Esse contexto foi repassado a cada uma das docentes, para ser utilizado com o OAC HF. Diante do contexto disponibilizado, foram necessárias alterações na interface do recurso de modo que o mesmo se adequasse ao planejamento.

Para a realização deste 2º momento foi levado em consideração somente a Atividade 1 do OAC HF, conforme apresentado na Figura 6.4. Essa atividade é formada por seis cenários.

Figura 6.4: Tela Atividade 1 do OAC Histórias Fantásticas

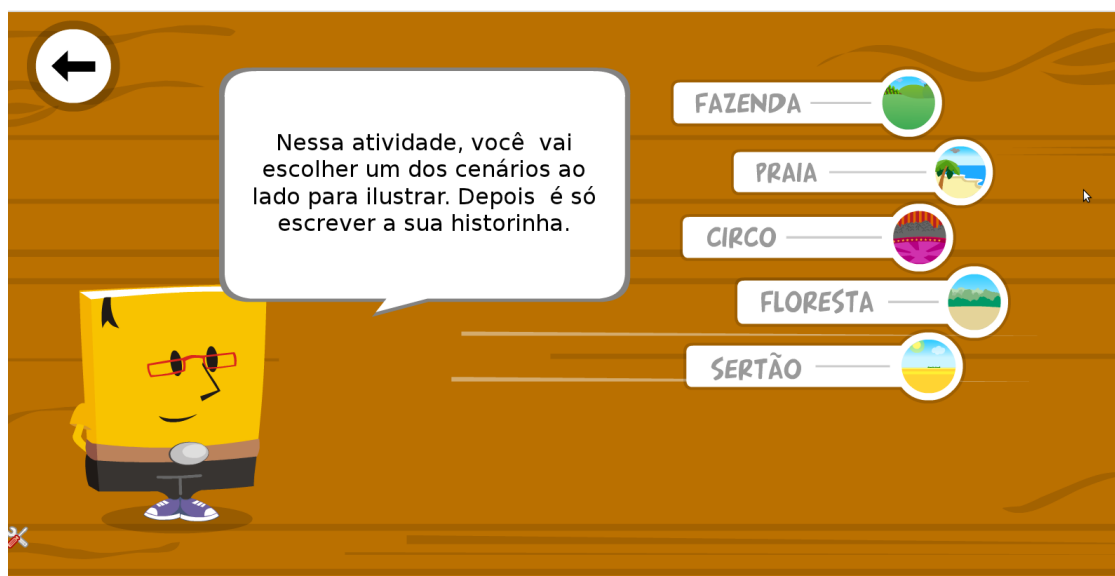


Fonte: Própria Autora

Como observado no planejamento dessa avaliação, o contexto utilizado pelos docentes envolvem quatro situações a serem realizadas no cenário FAZENDA. A descrição do contexto, repassada aos professores, foi realizada da seguinte forma:

**Professor (a), suponha que você necessite trabalhar a produção textual com seus alunos. O gênero textual escolhido será uma fábula e, para isso, você contará com a ajuda do OAC Histórias Fantásticas. Com base nessas**

Figura 6.5: Tela Atividade 1 do OAC Histórias Fantásticas (Customização 1)



Fonte: Própria Autora

**informações, analise cada uma das situações fornecidas e indique a(s) ação (ões) necessária(s) para adaptar o OA a esse contexto.**

Considerando esse contexto, foram propostas quatro situações que serviram de estímulo para a utilização das ações de customização pelas docentes.

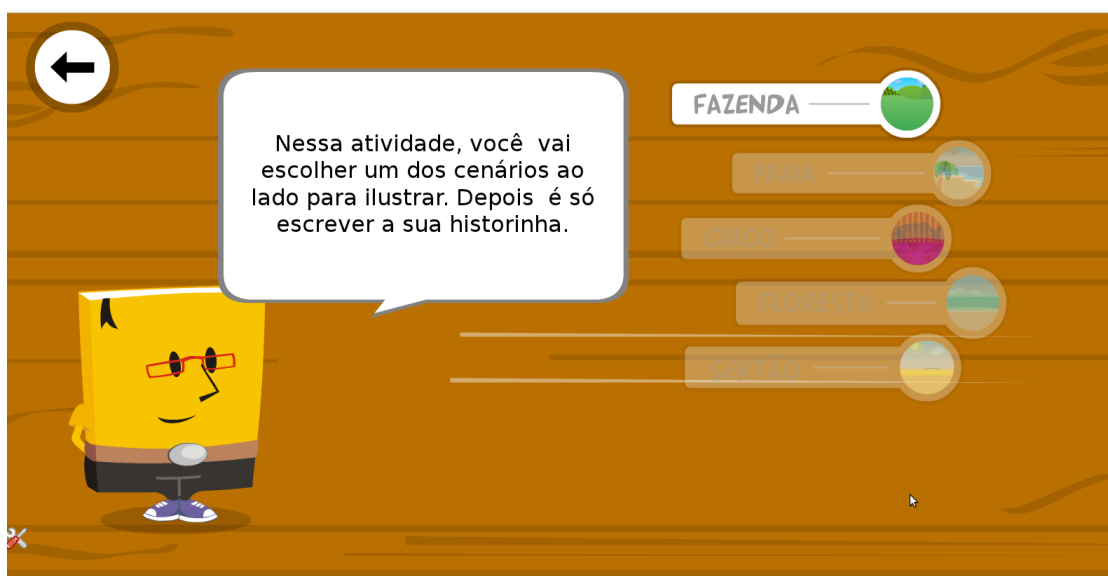
**Situação 1:** No OAC Histórias Fantásticas você dispõe de seis cenários. Qual deles não deve ser apresentado aos alunos para criação da fábula, uma vez que seu contexto não permite a produção deste tipo de gênero textual?

Nesta situação, a ação de customização a ser utilizada é o **Esconder**, isto porque essa ação, conforme descrita no Capítulo 3, permite que o elemento seja retirado da interface, fazendo com que ele não esteja no campo de visualização do usuário. Deve-se fazer uso dessa ação, sempre que for necessário não repassar o conhecimento, ao usuário, sobre determinado elemento ou atividade que não tenha nenhuma relação com o que foi planejado. Sendo assim, a intervenção correta para solucionar a problemática da situação apresentada é **Esconder** o cenário ESCOLA (Figura 6.5).

**Situação 2:** Utilize o cenário FAZENDA para explicar o gênero fábula aos alunos. Que ação deve ser executada com os outros quatro cenários para que você consiga centrar a atenção dos alunos somente no cenário indicado?

Para solucionar a problemática da situação apresentada convém fazer uso da ação de **Desabilitar**. Essa ação possibilita que o usuário tenha conhecimento das atividades, cenário ou elementos presentes no recurso, mas não tenha acesso as mesmas. O objetivo dessa ação é dar condição ao docente de executar seu planejamento sem sofrer interferências quer seja por outra atividade ou cenário, restringindo o acesso do usuário somente ao cenário que deve ser explorado. Sendo assim, para atingir o objetivo apresentado nesta situação, convém **Desabilitar** cenários PRAIA, CIRCO, FLORESTA e SERTÃO (Figura 6.6).

Figura 6.6: Tela Atividade 1 do OAC Histórias Fantásticas (Customização 2)



Fonte: Própria Autora

**Situação 3:** No cenário Fazenda, você dispõe de botões contendo personagens, objetos e animais. Se você irá propor a criação de uma fábula a seus alunos, que mudanças necessitarão ser feitas nestes botões?

Em todos os cenários do OAC Histórias Fantásticas, conforme apresentado na Figura 6.7, existem três botões indicando os personagens, os objetos e os animais que podem ser utilizados em cada um deles. Levando em consideração o contexto proposto, é importante observar que o gênero fábula é caracterizado por ser uma narrativa figurada, em que os personagens são geralmente animais e que estes possuem características humanas. Sendo assim, a mudança necessária a ser realizada no recurso deve possibilitar que duas ações sejam executadas, ou seja, que o botão PERSONAGENS,

e conseqüentemente seus elementos, sejam **Escondidos** e que o botão ANIMAIS seja **Substituído** pelo botão PERSONAGENS, fazendo com que os animais utilizados no cenário FAZENDA sejam agora os personagens do cenário (Figura 6.8)

Figura 6.7: Tela do Cenário FAZENDA do OAC Histórias Fantásticas



Fonte: Própria Autora

**Situação 4:** Com a retirada de um dos botões do cenário fazenda, que tipo de ajuste será necessário fazer no botão AJUDA?

Com as intervenções nos botões Personagens e Animais sugere-se que intervenções sejam realizadas no texto de AJUDA, conforme apresentado na Figura 6.9. Para esta situação, é necessário que o texto seja editado buscando excluir o nome do botão animais do texto. Dessa forma, a ação de **Editar** o texto é a indicada para solucionar a problemática apresentada nesta situação (Figura 6.10).

**3ª etapa:** Concluídas as adaptações necessárias no OAC Histórias Fantásticas para a produção de fábulas, foram realizadas entrevistas com as docentes que serviram para capturar as impressões das mesmas sobre as ações de customização e a relevância do OAC na prática docente. Os resultados dessa avaliação estão descritos a seguir.



Figura 6.8: Tela Customizada do Cenário FAZENDA do OAC Histórias Fantásticas



Fonte: Própria Autora

### 6.3.2.2 Resultados Alcançados

Da mesma forma como realizado na avaliação na concepção e desenvolvimento, foram realizadas gravações nas quais foram registradas as impressões dos avaliados. Adicionadas às informações registradas nos Diários de Campo pela moderadora, esses dados também foram organizadas em tópicos e analisados individualmente. A seguir são apresentados os tópicos utilizados, assim como as transcrições dos trechos das gravações e as impressões da moderadora sobre os mesmo.

#### **Tópico:** Necessidade de Customizar

A customização guiada possibilita a realização de diversas ações de adaptação em um OA. Essas ações devem ser realizadas pelos professores para responder suas necessidades pessoais em termos de funcionalidades desse recurso. As informações sobre as necessidade de customização foram levantadas da seguinte forma:

[DA] “Se chegar pra criança de hoje e falar que está errado, eles dizem: mas porque está errado?”

[DB] “As dificuldades dos alunos está em ler as coisas do OA. O que a atividade pede.”

Figura 6.9: Tela do Cenário FAZENDA com o conteúdo do botão AJUDA



Fonte: Própria Autora

[DB] “Difícilmente um recurso tem exatamente o que a gente precisa naquele momento.”

[DC] “os motivos que levam a não utilização de recursos, mesmo frente à escassez dos mesmos, estão relacionados às questões epistemológica e questões do *feedback*. Isto implica dizer que os recursos não possuem *feedback* adequado, pois atualmente, os alunos questionam o porquê do erro, não aceitando receber somente o rótulo de erro.”

**Impressões da Moderadora:** a necessidade por customização é bastante clara. As docentes apresentaram argumentos que indicam que pequenas modificações podem suplantar as limitações de adequação de um recurso a ser utilizado.

**Tópico:** Impactos da Customização

A realização das ações de customização possibilitam a modificação de vários aspectos de um OA. Sobre os impactos dessas ações foram levantados as seguintes informações:

[DA] “O OAC não é mais difícil do que o OA monolítico, mas o professor precisa estar atento ao que customizar.”

Figura 6.10: Tela Customizada do Cenário FAZENDA com o conteúdo do botão AJUDA



Fonte: Própria Autora

[DA] “Os ajustes são fantásticos, porque com eles se consegue atingir qualquer aluno.”

[DB] “Os ajustes permitidos no OAC são importantes para o professor, porque ele pode mudar. Não é algo pronto. Por mais que possam ter vários OA existem peculiaridades que o professor não encontra pronto.”

[DC] “As ações são bem intuitivas.”

**Impressões da Moderadora:** a flexibilidade dos OAC podem ser observadas nas colocações das docentes. Pode ser observado que as ações de customização foram facilmente compreendidas e, portanto, empolgaram as docentes com as novas possibilidades abertas por essas ações. Contudo, é importante controlar o que pode ser feito pelo professor, pois essas ações podem impactar na qualidade do recurso.

**Tópico:** Impacto para o Professor

O professor é o ator principal das ações de customização. Ele é quem deve lançar mão dessas ações para realizar as modificações nos recursos. Portanto, os impactos maiores dessas ações são experimentados por ele. Sobre essas impactos para os professores foram levantadas as seguintes informações:

[DA] “O OAC interfere na autonomia docente, com certeza.”

[DB] “... agora eu consigo ajustar o recurso às minhas necessidades. Não preciso ficar pedindo para a turma desconsiderar determinadas informações do recurso.”

[DC] “é importante mostrá-los [professores] que é possível ter bons recursos no computador e mais ainda, de fazê-los perceber a importância de inserir o computador dentro do planejamento de sua aula. Com a possibilidade de adaptar o recurso isso fica muito mais fácil, pois depende exclusivamente de nós professores.”

**Impressões da Moderadora:** é possível perceber que as docentes descreveram a importância dessa possibilidade de adaptar um OA na adoção desse tipo de recurso pelos professores. Além disso, com essas ações direcionadas aos professores, os mesmos tornam-se independentes para adaptar os recursos, sendo responsáveis diretos pelos resultados dessas adaptações.

## 6.4 Discussão

A utilização de uma abordagem empírica qualitativa para a avaliação das contribuições principais desta tese demonstraram o que já era esperado para este tipo de avaliação. Segundo Taylor e Bogdan (1998) as avaliações realizadas utilizando a abordagem qualitativa levantam indícios sobre aspectos que dificilmente podem ser precisados por outros métodos de avaliação. Esses aspectos são relacionados à comportamentos, ações, interesses dentre outros, associados à “problemas humanos” (SEAMAN, 1999).

Tendo essa premissa em mente, as avaliações realizadas neste capítulo foram planejadas e executadas com a intenção de perceber os impactos das pesquisas realizadas nesta tese no dia-a-dia dos potenciais usuários dos resultados dessas pesquisas.

Seguindo um protocolo de avaliação definido previamente, e constituindo grupos focais para aplicar o planejamento realizado, foram avaliados tanto as equipes de produção de OAC como os professores.

As equipes de produção de OAC são usuárias do CLO *Studio*. Essa ferramenta de autoria, conforme levantado na avaliação realizada, inova ao fornecer diversas funcionalidades para essas equipes, tais como simulação e reuso de componentes. Além disso, O CLO *Studio* também implementa os conceitos de customização guiada, fato que estimula a participação efetiva de uma equipe pedagógica na concepção dos OAC.

Os OAC tem como usuários principais os professores. Esses atores são os mais afetados com as ações de customização inseridas nos OA. Essa ações por um lado abrem um conjunto de possibilidades para esses profissionais, que terão flexibilidade para adaptar os recursos às suas necessidades pontuais, ampliando sua autonomia da utilização desse tipo de recurso. Por outro lado, aumentam a responsabilidade desse profissional com a qualidade do que está sendo modificado, pois essa autonomia adquirida deve ser utilizada com bom senso. Contudo, esse problema pode ser tratado pela customização guiada associada ao grau de liberdade do professor que irá utilizar o recurso, visto que essas modificações podem ser controladas até certo nível.

Retomando as hipóteses levantadas nas seções 6.2.2.1 e 6.2.3.1, é possível compreender a preocupação da avaliação do usuário frente as propostas apresentadas nesta tese. Essas hipóteses tratam de questões como eficiência de mecanismos, compreensão modelos, autonomia do professor, dentre outros aspectos. Considerando as análises realizadas na execução da avaliação da concepção e desenvolvimento e da utilização de OAC, o Quadro 6.5 apresenta as evidências que comprovam essas hipóteses.

## 6.5 Conclusão

Diferente das demais ferramentas de autoria discutidas no Capítulo 2 desta tese, a utilização da CLO *Studio* destina-se a uma equipe de desenvolvimento de OAC. Isto porque sua filosofia é focada na construção do projeto instrucional que favorece a construção do projeto gráfico e sua consequente implementação. Para tanto, o CLO *Studio* tem duas grandes preocupações. A primeira delas é criar mecanismos que favorecessem as interações entre os participantes das equipes de produção de OAC, de modo que as interações estabelecidas resultassem em claras interpretações do que

Quadro 6.5: Avaliação das Hipóteses

ID	Hipótese/Sub-hipótese	Resultado/Evidência
$H_A$	<b>A Engenharia de Software Orientada a Modelos possibilita uma melhoria na comunicação entre os membros da equipe de produção, favorecendo um melhor tratamento dos aspectos de customização guiada nestes recursos.</b>	Aceita. A comunicação foi realizada de modo dinâmico e o compartilhamento de ações foi incentivado naturalmente pelo CLO <i>Studio</i> . Além de estimular a colaboração entre os membros durante o processo de desenvolvimento.
$H_1$	O modelo de arquitetura construído através da DSL é compreendido por todos os tipos de atores do processo.	Aceita. Os conceitos manipulados foram facilmente assimilados por todos os membros da equipe.
$H_2$	O CLO <i>Studio</i> é de fácil manipulação por todos os tipos de atores do processo.	Aceita com restrições. A representação de subcenas pode tornar a compreensão de alguns diagramas complexos. Contudo, os usuários que adquirem alguma experiência não sentem dificuldades com esse conceito.
$H_B$	<b>A utilização de recursos adaptáveis (OA customizáveis) em sala aula eleva a autonomia dos professores na criação de novas oportunidades e estratégias de ensino.</b>	Aceita. O processo de construção da Fábula poderia ser iniciado de diversas maneiras. Quem define a forma é o professor.
$H_3$	Os elementos disponíveis para acessar as ações de customização na interface de um OAC são intuitivos e de fácil manipulação.	Aceita. Os elementos foram utilizados de modo direto pelos professores sem a necessidade da intervenção da moderadora.
$H_4$	As ações de customização e suas respectivas consequências podem ser facilmente compreendidas.	Aceita. As ações de customização são visualizadas diretamente pelo professor na interface do recurso.
$H_5$	Novos recursos podem ser criados a partir da customização de um OAC para responder por novas necessidades específicas.	Aceita. Com a compreensão das ações de customização, passam a surgir, para os professores, novas ideias para modificar o OAC.

Fonte: Própria Autora

foi especificado. Já a segunda, diz respeito ao recurso produzido por intermédio desta ferramenta, mas especificamente, aos ganhos que o uso de um OA customizável pode proporcionar à prática docente.

Com base nos mecanismos de interação, o CLO *Studio* gera documentos que servem para nortear tanto quem projeta graficamente o recurso, quanto quem o implementa, permitindo, dessa forma, a geração automática do OA customizável (OAC). Outra característica do OA projetado pelo CLO *Studio*, é que ele permite delimitar, ainda na fase de projeto, o tipo de intervenção que cada elemento do OA pode sofrer. Isto só é possível, porque o CLO *Studio* dá suporte a ideia de customização guiada, proposta nesta tese.

Como apresentado na avaliação realizada para a fase de concepção e desenvolvimento com o CLO *Studio*, esses objetivos foram alcançados, visto que as evidências levantadas mostram que essa ferramenta se apresentou como um ponto de colaboração na qual os membros das equipes puderam interagir e trocar opiniões, harmonizando os interesses e evitando retrabalho.

No que diz respeito aos possíveis ganhos acarretados com a utilização dos OAC pelos professores, a autonomia desse profissional é o fator mais relevante. Isto porque mais do que possibilitar a adaptação de OAC, a preocupação da customização guiada é possibilitar a criação de recursos flexíveis o suficiente para incentivar esses profissionais a ampliar a utilização desses recursos e a direcioná-los às suas necessidades. Neste sentido, acredita-se que a avaliação realizada sobre a utilização dos OAC aponta para um ganho relacionado à autonomia, visto que esses professores podem lançar mão das ações de customização para criar novas possibilidades pedagógicas com seus recursos.

É importante ressaltar, que mesmo diante dos ganhos obtidos tanto pelas equipes de produção de OAC como pelos professores, ainda é necessário sensibilizar todos esses profissionais para os benefícios da customização guiada e para as vantagens associadas a liberdade adquirida por esses profissionais através desse conceito.

## Capítulo 7

### Conclusão e Trabalhos Futuros

“Valeu a pena? Tudo vale a pena Se a alma não é pequena.  
Quem quer passar além do Bojador tem que passar além  
da dor.”  
– *Fernando Pessoa*

Diante do conceito de customização guiada e da sua aplicação para a produção de OA, do processo de desenvolvimento CLassROOM e da ferramenta de autoria CLO Studio propostos nesta tese, são apresentados neste capítulo as conclusões dos principais pontos relacionados a essa pesquisa. Além disso, são apontadas diversas possibilidades para a ampliação dos resultados deste trabalho através da indicação de trabalhos futuros.

#### 7.1 Conclusão

A estratégia de Customização Guiada, apresentada nesta tese, tem por finalidade permitir que professores possam se apropriar de RED do tipo OA e adaptá-los às suas necessidades. Neste contexto, o termo "Customização" diz respeito às ações que poderão ser realizadas diretamente nas interfaces desses recursos de modo a produzir novas versões dos mesmos. Essas versões, por sua vez, funcionarão como novos recursos com características específicas para serem utilizados também em situações específicas. Dessa forma, a partir de um único OA, poderiam ser geradas diversas versões diferentes, cada uma adaptada a um contexto e necessidade distintas. Assim, o professor consegue atingir a autonomia na utilização desse recurso, visto que, com essa abertura fornecida pela customização, o recurso pode também ser adaptado ao planejamento docente.



Já o termo "Guiada" diz respeito à importância de considerar os mais diversos níveis de conhecimento desses professores para realizar essas adaptações. Assim, a criação de recursos com características de customização deve ser realizada por uma equipe de produção que precisa decidir, em conjunto, exatamente que ações de customização serão "liberadas" para serem utilizadas no recurso. Essas decisões sobre o que disponibilizar para ser modificado é que guiará as ações dos professores.

Sendo assim, a equipe de produção deve trabalhar de modo integrado e coeso, de forma a construir um recurso que possa atender às mais diferentes necessidades. Essa integração exige a utilização de mecanismos que incrementem a comunicação entre os diferentes profissionais dessa equipe, para que cada um possa participar de forma efetiva nas decisões, visto que essas decisões afetam diretamente a qualidade do produto final. Dessa forma, a customização guiada é uma estratégia que, mesmo tendo como objetivo maior ampliar o nível de abertura do recurso para o professor, também foca efetivamente em um processo de produção e em uma ferramenta para a criação desses recursos, haja vista que é na produção que os aspectos de customização são inseridos no contexto do mesmos.

Ao longo do desenvolvimento desse trabalho, essas situações foram tratadas de modo pontual. Inicialmente, para que um recurso possa seguir os preceitos da customização guiada, essa abordagem foi caracterizada através de alguns conceitos. O primeiro deles diz respeito a ideia de Potencial Pedagógico dos elementos que compõem a interface de um recurso. Esse conceito está ligado à importância que a equipe de produção deve considerar para associar ações de customização a determinados elementos de um recurso. Em seguida, com base na indicação do potencial pedagógico dos elementos, foram definidos os Graus de Liberdade associados ao professor. Esse conceito indica que nível de maturidade um professor tem para adaptar ou modificar um recurso sem interferir em suas premissas pedagógicas. Dessa forma, é possível afirmar que as ações de customização são associadas aos graus de liberdade ou seja, quanto maior o grau de liberdade do professor maior será a permissão e conseqüentemente o acesso desse docente a uma quantidade de ações disponíveis para adaptar o recurso. Essa relação entre potencial pedagógico e grau de liberdade indica, mais uma vez, que é na produção do recurso que as possibilidades de customização são definidas.

Devido à importância da produção dos recursos na definição das estratégias de customização guiada, foi necessária a criação de um processo de produção de recursos customizáveis. Esse processo, denominado CLAssROOM, utiliza uma estratégia orientada a modelos para a criação de Objetos e Aprendizagem Customizáveis (OAC). Pelo fato desse processo utilizar modelos como artefatos principais, ele se caracteriza pela integração das ações de produção em um maior nível de abstração. Essa abstração, implementada através de um metamodelo, é manipulada pela equipe de produção através de um conjunto de elementos que expressam as informações necessárias para conceber o recurso. Essas informações são direcionadas aos mais diversos profissionais da equipe de produção na forma de uma linguagem específica de domínio (DSL). Essa DSL, por sua vez, funciona como um idioma único para todos os profissionais, permitindo que todos possam trocar informações em um mesmo nível. Dessa forma, essa DSL unifica o canal de comunicação entre esses profissionais, o que propicia uma maior integração e, conseqüentemente, uma maior qualidade dos recursos desenvolvidos.

Como forma de auxiliar a produção dos recursos, também foi desenvolvido nesse trabalho uma ferramenta de autoria para automatizar as ações do processo CLAssROOM. Essa ferramenta, denominada de CLO *Studio*, faz uso da filosofia do wireframe, permitindo que a manipulação da arquitetura das interfaces dos recursos seja realizada diretamente pelos profissionais envolvidos. Como o CLO *Studio* dá suporte à DSL proposta neste trabalho, essa ferramenta serve de palco para integração da produção, visto que todos os profissionais são estimulados a participar da concepção dos recursos. Além disso, a possibilidade de realizar simulações do funcionamento do recurso oferecida pelo CLO *Studio*, permite a visualização dos resultados e a antecipação de discussões coletivas. Entretanto, para que possam ser preservadas as peculiaridades das tarefas de cada um dos profissionais envolvidos no processo de produção, o CLO *Studio* gera automaticamente um *storyboard* contendo as informações especificadas de modo coletivo para servir de base para os trabalhos das equipes individuais. Dessa forma, os profissionais de *design* poderão construir os elementos visuais do recurso baseado na especificação realizada, na qual ele próprio pode contribuir. Além disso, também são gerados automaticamente pelo CLO *Studio* um conjunto de *templates* de implementação na linguagem final na qual o recurso será codificado. Esses *templates* devem ser

utilizados pela equipe de desenvolvimento para realizar ajustes no funcionamento do recurso. Posteriormente, depois que tanto os profissionais de *desing* quanto os desenvolvedores finalizarem suas ações, os componentes visuais e os *templates*, devidamente ajustados, são acessados pelo CLO *Studio* para gerar o código final do recurso. Essa abordagem colaborativa utilizada pelo CLO *Studio*, facilita a interação entre os componentes da equipe e amplia a comunicação entre os diferentes profissionais, permitindo que diversas situações de conflito possam ser antecipadas e resolvidas mais rápida e precisamente.

Com a intenção de analisar os impactos do que foi proposto nesta tese, foram realizadas avaliações considerando os principais aspectos envolvidos tanto na concepção e desenvolvimento como na utilização de OAC. Utilizando uma estratégia de avaliação empírica qualitativa, foram avaliados a integração e comunicação entre os profissionais de produção de OAC e a autonomia adquirida pelos professores quando da utilização desse tipo de recurso. Os resultados levantados foram animadores e estimulantes, visto que as hipóteses levantadas na avaliação foram confirmadas e diversas outras sugestões foram dadas, indicando assim que ainda existem diversas formas de ampliar as ações relacionadas à customização guiada. Deste modo, na seção 7.2 são apresentados os trabalhos a serem realizados futuramente.

## 7.2 Trabalhos Futuros

As perspectivas de trabalhos futuros para esta tese podem ser agrupadas em três vertentes. A primeira diz respeito a exploração dos conceitos de customização guiada em outros tipos de RED. A segunda foca na realização de melhorias referentes a usabilidade do CLO *Studio* e a terceira trata de melhorias nos OAC referentes a produção, armazenamento, acesso, dentre outros.

Com relação à customização guiada, acredita-se que os mesmos benefícios analisados na utilização dessa estratégia nos OA possam ser conseguidos em outros tipos de RED, tais como páginas *Web*, softwares educativos, dentre outros. Dessa forma, uma pesquisa em aberto é a aplicação das contribuições desta tese em termos de processos

e ferramentas na perspectiva desses outros tipos de recursos.

Com relação ao CLO Studio, é possível listar os seguintes pontos a serem explorados:

- **Criação de outras DSL:** ampliar a diversidade de OA produzidos com o CLO *Studio* através da exploração de outros tipos de OA. Isto implica na inserção de novas notações no CLO *Studio* de modo a trabalhar com outros OA diferentes de simulação/animação.
- **Melhorar apresentação subcenas:** atualmente o CLO *Studio*, conforme validação apresentada no Capítulo 6, atende às necessidades da equipe multidisciplinar no que se refere à concepção e desenvolvimento de OAC simplificado. No entanto, quando o OAC a ser desenvolvido possuir muitas subcenas, o espaço destinado no CLO *Studio* para projetá-las não favorece uma clara visualização do que foi especificado. Para amenizar essa limitação, a ferramenta dá condições de associar à representação gráfica uma representação textual que fornecerá mais detalhes dos cenários de cada subcena. No entanto, devido à limitação da representação textual, é importante desenvolver na ferramenta condições de não somente utilizar as subcenas, mas acima de tudo visualizá-las e compreendê-las.
- **Análise formal das ações de customização:** após projetar um OAC no CLO *Studio* é possível realizar uma simulação do que fora projetado no intuito de detectar alguma inconsistência no projeto. No entanto, essas inconsistências podem ser provocadas pelo docente, caso as ações de customização não tenham sido bem delimitadas pela equipe pedagógica, haja vista que as mesmas não são, e nem devem ser, visualizadas na simulação, devido às inúmeras possibilidades que podem ser proporcionadas. Dessa forma, é válido ressaltar a importância de associar ao CLO *Studio* o LOCPN (SOUZA et al., 2007). O LOCPN (*Learning Objects production with Colored Petri Nets*) é um modelo formal desenvolvido especificamente para auxiliar o processo de desenvolvimento de OA capaz de validar as diversas situações possíveis que um cenário ou recurso venha a ter, de modo a fornecer uma visão do comportamento do recurso frente às inúmeras possibilidades proporcionadas pelas ações de customização, minimizando, assim,

as inconsistências do OAC. Assim, uma proposta seria a ampliação do LOCPN para que esse possa expressar formalmente as ações de customização e essas, por sua vez, sejam analisadas automaticamente pelo CLO *Studio*.

- **Incremento das ações de customização:** as ações de customização dão condições ao docente de aproximar ao máximo o OAC da situação e contexto necessários. Para ajudar nesse processo de aproximação entre recurso e aluno, convém incrementar as ações de customização para que elas favoreçam, por exemplo, a acessibilidade. Exemplo disso seria fragmentar os áudios ou mesmo inserir legendas nos trechos de vídeos fragmentados, caso o professor necessitasse desse recurso para atender ao seu público.
- **Produção de OAC para outros dispositivos:** em meio à diversidade de dispositivos desenvolvidos é de suma importância desenvolver OAC portáteis, até porque se a palavra de ordem defendida pelo OAC é liberdade, não é coerente desenvolver esse tipo de recurso somente para PC. Para isso, é necessário inserir outros tipos de eventos que atualmente no CLO *Studio* são realizados somente pelo *mouse*, mas que em um *tablet*, por exemplo, são realizados por gestos.

Já com relação aos OAC, é possível listar as seguintes pesquisas em aberto:

- **Controle de versões de OAC:** ao utilizar as ações de customização especificadas no OAC, o docente tem a possibilidade de modificá-lo. Para que essas modificações estejam disponíveis para utilização futura, atualmente o CLO *Studio* salva essas versões em um arquivo *XML* no servidor. Contudo, atualmente essas versões estão disponíveis para todos os usuários que tenham acesso ao servidor. Deste modo, é importante que seja implementado um mecanismo mais robusto de controle de versões que possibilite a associação das versões dos OAC com os seus respectivos “autores”, para que o acesso as mesmas seja corretamente controlado.
- **Repositório (servidor) de OAC:** um dos critérios utilizados para o desenvolvimento de um OAC é baseado na componentização das imagens. Sendo assim, para que haja uma maior agilidade na produção dos OAC através da reutiliza-

ção dos componentes, convém desenvolver um repositório de acesso livre para o armazenamento dessas imagens.

- **Explorar o conceito de potencial pedagógico dos componentes:** cada elemento que compõe um OAC recebe, durante a fase de projeto, um nível de importância, quer seja relacionado à navegabilidade do recurso ou a sua potencialidade pedagógica. Diante disso, é importante buscar meios para explorar de forma mais precisa as métricas aplicadas a esses elementos como forma de sinalizar a existência ou não de potencial pedagógico nos mesmos.
- **Acesso concorrente ao CLO Studio:** disponibilizar a ferramenta CLO *Studio* em rede de modo a favorecer a produção de OAC, em paralelo, por diversas equipes.

## REFERÊNCIAS

- ALAM, G.; ISMAIL, K.; MISHRA, P. Do developing countries need education laws to manage its system or are ethics and a market-driven approach sufficient? *Afr. J. Bus. Manage*, v. 4, n. 15, p. 3406–3416, 2010.
- ALBRIGHT, P. Open educational resources, open content for higher education: Final forum report. *International Institute for Educational Planning*, p. 16, 2005. Disponível em: <[http://portal.unesco.org/ci/en/files/21713-11438000259OER\\_Forum\\_Final\\_Report.pdf/OER%2BForum%2BFinal%2BReport.pdf](http://portal.unesco.org/ci/en/files/21713-11438000259OER_Forum_Final_Report.pdf/OER%2BForum%2BFinal%2BReport.pdf)>. Acesso em: 18 jan. 2012.
- ALMEIDA, P. de. O computador na escola: contextualizando a formação de professores. 2000.
- AMARAL, L.; GOMES, T.; SOUZA, M.; PEQUENO, M. et al. Um aprimoramento do modelo de processo de criação de objetos de aprendizagem do projeto rived. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2006. v. 1, n. 1.
- AMIEL, T.; OREY, M.; WEST, R. Recursos educacionais abertos (rea): modelos para localização e adaptação open educational resources (oer): models for adaptation and localization. *ETD-Educação Temática Digital-ISSN 1676-2592*, v. 12, n. mar., p. 112–125, 2010.
- AMIEL, T.; SQUIRES, J.; OREY, M. Four strategies for designing instruction for diverse cultures. *Educational Technology*, v. 49, n. 6, p. 28–34, 2009.
- APPLICATION, E. Grupos focais na investigação qualitativa da identidade organizacional: exemplo de aplicação. SciELO Brasil, 2004.
- ARTEAGA, J.; URRUTIA, B.; RODRÍGUEZ, F.; SALAS, P. Metodología para elaborar objetos de aprendizaje e integrarlos a un sistema de gestión de aprendizaje. 2006.
- ARTURO, B.; JAIME, M.; FRANCISCO, A.; MARIA, G. Developing large scale learning objects for software engineering process model through MIDOA model. In: IEEE. *Computer Science (ENC), 2009 Mexican International Conference on*. [S.l.], 2009. p. 203–208.

- BAEK, Y.; JUNG, J.; KIM, B. What makes teachers use technology in the classroom? exploring the factors affecting facilitation of technology with a korean sample. *Computers & Education*, Elsevier, v. 50, n. 1, p. 224–234, 2008.
- BAKI, A.; ÇAKIROGLU, Ü. Learning objects in high school mathematics classrooms: Implementation and evaluation. *Computers & Education*, Elsevier, v. 55, n. 4, p. 1459–1469, 2010.
- BALDA, J.; LÓPEZ, M. Locomo: Metodología de construcción de objetos de aprendizaje. 2008.
- BARAJAS, A.; MUÑOZ, J.; ÁLVAREZ, F. Modelo instruccional para el diseño de objetos de aprendizaje: Modelo midoa. *Ponencia presentada en Virtual Educa*, 2007.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 2003.
- BATSON, T.; PAHARIA, N.; KUMAR, M. A harvest too large?: A framework for educational abundance. *Opening Up Education: The Collective Advancement of Education through Open Technology, Open Content, and Open Knowledge*, p. 89–103, 2008.
- BECERRA J. R.; SIQUEIRA, F. L. B. B. Modelo de fábrica de software. In: *VII Simpósio Internacional de Melhoria de Processos de Software. São Paulo*. [s.n.], 2005. Disponível em: <<http://www.rived.mec.gov.br/artigos/rived.pdf>>. Acesso em: 18 jan. 2012.
- BOND, S.; INGRAM, C.; RYAN, S. Reuse, repurposing and learning design-lessons from the dart project. *Computers & Education*, Elsevier, v. 50, n. 2, p. 601–612, 2008.
- BUDINSKY, F. *Eclipse modeling framework: a developer's guide*. [S.l.]: Addison-Wesley Professional, 2004.
- CANAL, C.; MURILLO, J.; POIZAT, P. et al. Software adaptation. *L'objet*, v. 12, n. 1, p. 9–31, 2006.
- COMBEMALE, B.; CRÉGUT, X.; GIACOMETTI, J.; MICHEL, P.; PANTEL, M. et al. Introducing simulation and model animation in the mde topcased toolkit. 2008.
- CONER, A. Personalization and customization in financial portals. *Journal of American Academy of Business*, v. 2, n. 2, p. 498–504, 2003.
- CONTE, T.; CABRAL, R.; TRAVASSOS, G. Aplicando grounded theory na análise qualitativa de um estudo de observação em engenharia de software—um relato



- de experiência. In: *V Workshop "Um Olhar Sociotécnico sobre a Engenharia de Software" (WOSES 2009)*. [S.l.: s.n.], 2009. p. 26–37.
- CORDEIRO, R.; RAPKIEWICZ, C.; CANELA, M.; SANTOS, A.; CARNEIRO, E. Utilizando mapas conceitual, de cenário e navegacional no apoio ao processo de desenvolvimento de objetos de aprendizagem. *CINTED-UFRGS–Novas Tecnologias da Educação*, v. 5, n. 1, 2007.
- CRESWELL, J. *Educational research: Planning, conducting, and evaluating quantitative and*. 2002.
- DAGIENE, V.; ZILINSKIENE, I. Localization of learning objects in mathematics. 10th int. In: *Conference: Models in Developing Mathematics Education. Dresden: The University of Applied Sciences (FH)*. [S.l.: s.n.], 2009. p. 129–133.
- D'ANJOU, J. *The Java developer's guide to Eclipse*. [S.l.]: Addison-Wesley Professional, 2005.
- DICTIONARY, O. *Oxford: Oxford University Press*. 1989.
- DJURIC, D.; JOVANOVIC, J.; DEVADZIC, V.; SENDELJ, R. Modeling ontologies as executable domain specific languages. In: *ACM. Proceedings of the 3rd India software engineering conference*. [S.l.], 2010. p. 83–92.
- DONOVAN, M.; BRANSFORD, J.; PELLEGRINO, J. *How people learn: Brain, mind, experience, and school*. [S.l.]: Washington, DC: National Academy Press, 2000.
- DUTRA, R.; TAROUÇO, L. *Objetos de aprendizagem: uma comparação entre scorm e ims learning design*. 2011.
- EASTERBROOK, S.; SINGER, J.; STOREY, M.; DAMIAN, D. Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, Springer, p. 285–311, 2008.
- ERTMER, P.; OTTENBREIT-LEFTWICH, A. Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. *Journal of Research on Technology in Education*, International Society for Technology in Education, v. 42, n. 3, p. 255–284, 2010.
- FERNANDES, E.; COSTA, R. *Computadores: janelas para o mundo*. 2009. Revista Nova Escola. Disponível em: <<http://revistaescola.abril.com.br/edicoes-especiais-/029.shtml>>. Acesso em: 1 ago. 2011.

- FLICK, U.; KARDORFF, E. von; STEINKE, I. 1 what is qualitative research? an introduction to the field. *A Companion to*, p. 1, 2004.
- FOWLER, M.; PARSONS, R. *Domain-specific languages*. [S.l.]: Addison-Wesley Professional, 2010.
- FRAKES, W.; KANG, K. Software reuse research: Status and future. *Software Engineering, IEEE Transactions on*, IEEE, v. 31, n. 7, p. 529–536, 2005.
- FREIRE, P. Pedagogia da autonomia: saberes necessários à prática pedagógica. *São Paulo: Paz e Terra*, p. 165, 1996.
- FREUND, R. Mass customization, personalization and contextualized multiple competencies. In: *3rd International Conference on Mass Customization and Open Innovation*. [S.l.: s.n.], 2008. v. 3, n. 06.
- GAUTHIER, C.; TARDIF, M. *A pedagogia: teorias e práticas da antiguidade aos nossos dias*. [S.l.]: Vozes, 2010.
- GKATZIDOU, S.; PEARSON, E. The potential for adaptable accessible learning objects: A case study in accessible vodcasting. *Australasian Society for Computers in Learning in Tertiary Education*, 2009.
- GODOY, A. Introdução à pesquisa qualitativa e suas possibilidades. *Revista de administração de empresas*, São Paulo, v. 35, n. 2, p. 57–63, 1995.
- GRIFFITHS, D.; BEAUVOIR, P.; BAXENDALE, M. B.; HAZLEWOOD, P.; ODDIE, A. Development and evaluation of the reload learning design editor. 2007.
- GUERRERO, M.; GAONA, A.; RODRÍGUEZ, F. Construcción de objetos de aprendizaje de pruebas unitarias de la ingeniería de software a través de una metodología ligera. *Mexican International Conference on Computer Science. V Taller sobre Tecnología de Objetos de Aprendizaje (TaTOAje)*, 2007.
- GUNN, C.; WOODGATE, S.; O'GRADY, W. Repurposing learning objects: a sustainable alternative? *ALT-J*, Taylor & Francis, v. 13, n. 3, p. 189–200, 2005.
- HALLAK, J. Investing in the future: Seeing educational priorities in the developing world. *UNESCO, Paris*, 1999.
- HARMAN, K.; KOOHANG, A. *Learning objects: standards, metadata, repositories & LCMS*. [S.l.]: Informing Science, 2007.

- HATAKKA, M. Build it and they will come?—inhibiting factors for reuse of open content in developing countries. *The Electronic Journal of Information Systems in Developing Countries*, v. 37, n. 0, 2009.
- HEN-TOV, A.; LORENZ, D.; SCHACHTER, L. Modeltalk: A framework for developing domain specific executable models. *Arxiv preprint arXiv:0906.3423*, 2009.
- HICKSON, I.; HYATT, D. Html 5. *Working draft, W3C*, 2008.
- HOLZINGER, A.; EBNER, M. Interaction and usability of simulations & animations: A case study of the flash technology. In: *Proceedings of Interact.* [S.l.: s.n.], 2003. p. 777–780.
- HYLÉN, J. Open educational resources: Opportunities and challenges. *Proceedings of Open Education*, p. 49–63, 2006.
- IEEE. Draft standard for learning technology: Extensible markup language (xml) schema definition language binding for learning object metadata. *IEEE P1484.12.3, Draft 8*, 2005. Disponível em: <<http://www.msglobal.org/metadata/>>. Acesso em: 18 jan. 2012.
- INAN, F.; LOWTHER, D. Factors affecting technology integration in k-12 classrooms: a path model. *Educational Technology Research and Development*, Springer, v. 58, n. 2, p. 137–154, 2010.
- INITIATIVE, O. The open source definition. *URL: <http://www.opensource.org/docs/osd>*, accessed on, p. 10–17, 2007.
- IRELAND, C. Qualitative methods: From boring to brilliant. *Design Research: Methods and Perspectives*, p. 22, 2003.
- JACOBSON, M.; WELLER, M. A profile of computer use among the university of illinois humanities faculty. *Journal of Educational Technology Systems*, 1988.
- KAHLE, D. Designing open educational technology. *Opening up education: The collective advancement of education through open technology, open content, and open knowledge*, p. 27–45, 2008.
- KAPLAN, B.; MAXWELL, J. Qualitative research methods for evaluating computer information systems. *Evaluating the Organizational Impact of Healthcare Information Systems*, Springer, p. 30–55, 2005.
- KARLSCH, M. *A model-driven framework for domain specific languages demonstrated on a test automation language*. Tese (Doutorado) — Master thesis, Hasso-Plattner-Institute of Software Systems Engineering, Potsdam, Germany, 2007.

- KELLY, S.; TOLVANEN, J. *Domain-specific modeling: enabling full code generation*. [S.l.]: Wiley-IEEE Computer Society Pr, 2008.
- KOOHANG, A.; FLOYD, K.; STEWART, C. Design of an open source learning objects authoring tool—the lo creator. *Interdisciplinary Journal of E-Learning and Learning Objects*, v. 7, 2011.
- KORRES, M.; GARCÍA-BARRIOCANAL, E. Development of personalized learning objects for training adult educators of special groups. *Journal of Knowledge Management*, Emerald Group Publishing Limited, v. 12, n. 6, p. 89–101, 2008.
- KRUEGER, R.; CASEY, M. *Focus groups: A practical guide for applied research*. [S.l.]: Sage, 2009.
- LAWLESS, K.; PELLEGRINO, J. Professional development in integrating technology into teaching and learning: Knowns, unknowns, and ways to pursue better questions and answers. *Review of Educational Research*, Sage Publications, v. 77, n. 4, p. 575–614, 2007.
- LONGMIRE, W. A primer on learning objects. *Learning Circuits*, v. 1, n. 3, 2000.
- MACK, N.; WOODSONG, C.; MACQUEEN, K.; GUEST, G.; NAMEY, E. Qualitative research methods overview. *N. Mack, C. Woodsong, KM Macqueen, G. Guest, & E Namey*, p. 1–12, 2005.
- MEANS, B.; PLANNING, E. United States. Dept. of Education. Office of; DEVELOPMENT, P. *Implementing data-informed decision making in schools: Teacher access, supports and use*. [S.l.]: US Dept. of Education, Office of Planning, Evaluation and Policy Development, 2009.
- MEHLINGER, H.; POWERS, S. *Technology & teacher education: A guide for educators and policymakers*. [S.l.]: Houghton Mifflin, 2002.
- MERKS, E.; GROSE, R. *Eclipse Modeling Framework*. [S.l.]: Addison-Wesley Professional, 2003.
- MERNIK, M.; HEERING, J.; SLOANE, A. When and how to develop domain-specific languages. *ACM Computing Surveys (CSUR)*, ACM, v. 37, n. 4, p. 316–344, 2005.
- MONTEIRO, B.; CRUZ, H.; ANDRADE, M.; GOUVEIA, T.; TAVARES, R.; ANJOS, L. Metodologia de desenvolvimento de objetos de aprendizagem com foco na aprendizagem significativa. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2006. v. 1, n. 1, p. 388–397.

- MORSE, J. The significance of saturation. *Qualitative health research*, SAGE Publications, v. 5, n. 2, p. 147, 1995.
- MUELLER, J.; WOOD, E.; WILLOUGHBY, T.; ROSS, C.; SPECHT, J. Identifying discriminating variables between teachers who fully integrate computers and teachers with limited integration. *Computers & Education*, Elsevier, v. 51, n. 4, p. 1523–1537, 2008.
- MUSTARO, P.; SILVEIRA, I.; OMAR, N.; STUMP, S. Structure of storyboard for interactive learning objects development. *Learning objects and instructional design*, Informing Science, v. 3, p. 253, 2007.
- MYERS, M.; AVISON, D. Qualitative research in information systems. *Management Information Systems Quarterly*, MIS RESEARCH CENTER-SCHOOL OF MANAGEMENT, v. 21, p. 241–242, 1997.
- NASCIMENTO, A.; MORGADO, E. Um projeto de colaboração internacional na américa latina. 2003. Disponível em: <<http://www.rived.mec.gov.br/artigos/rived-.pdf>>. Acesso em: 18 jan. 2012.
- NETDAY. *The Internet, Technology and Teachers*. 2001. NetDay Survey 2001. Disponível em: <[http://www.netday.org/anniversary\\_survey.htm](http://www.netday.org/anniversary_survey.htm)>. Acesso em: 9 out. 2011.
- NEVES, J. Pesquisa qualitativa: características, usos e possibilidades. *Caderno de pesquisas em administração, São Paulo*, v. 1, n. 3, p. 2, 1996.
- NGUYEN, T.; BRUILLARD, É.; CACHAN-INRP, U.; UNIVERSUD, F. Exchanging digital educational resources among teachers: a survey in vietnam. In: *Proceedings of Society for Information Technology & Teacher Education International Conference 2011*. [S.l.: s.n.], 2011. p. 1592–1599.
- NICOLEIT, G.; PELEGRIN, D.; SOUZA, G.; ZANETTE, E.; SANTOS, C.; FIUZA, P. Planejamento e desenvolvimento do objeto de aprendizagem "regulação da liberação dos hormônios sexuais masculinos-rlhsmi". *RENOTE*, v. 4, n. 2, 2010.
- NIELSEN, J. Customization of uis and products. 2009. Disponível em: <<http://www.useit.com/alertbox/customization.html>>. Acesso em: 18 jan. 2012.
- OCHOA, X.; DUVAL, E. Measuring learning object reuse. *Times of Convergence. Technologies Across Learning Contexts*, Springer, p. 322–325, 2008.
- ONWUEGBUZIE, A.; COLLINS, K. A typology of mixed methods sampling designs in social science research. *The Qualitative Report*, v. 12, n. 2, p. 281–316, 2007.

- ONWUEGBUZIE, A.; LEECH, N. Sampling designs in qualitative research: Making the sampling process more public. *The Qualitative Report*, v. 12, n. 2, p. 238–254, 2007.
- PALAZZO, L. Sistemas de hipermídia adaptativa. In: *Anais do XXII Congresso da Sociedade Brasileira de Computação (SBC), Florianópolis*. [S.l.: s.n.], 2002.
- PATRICK, S. Ict in educational policy in the north american region. *International Handbook of Information Technology in Primary and Secondary Education*, Springer, p. 1109–1117, 2008.
- PELGRUM, W. Obstacles to the integration of ict in education: results from a worldwide educational assessment. *Computers & Education*, Elsevier, v. 37, n. 2, p. 163–178, 2001.
- PERRY, D.; PORTER, A.; VOTTA, L. Empirical studies of software engineering: a roadmap. In: ACM. *Proceedings of the Conference on the Future of Software Engineering*. [S.l.], 2000. p. 345–355.
- PERRY, D.; WOLF, A. Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 17, n. 4, p. 40–52, 1992.
- PESSOA, M.; BENITTI, F. Proposta de um processo para produção de objetos de aprendizagem. *HÍFEN*, v. 32, n. 62, 2008.
- PETRONI, A.; SOUZA, V. de. Vigotski e paulo freire: contribuições para a autonomia do professor. *Revista Diálogo Educacional, Curitiba*, v. 9, n. 27, p. 351–361, 2009.
- PHILIP, L. Combining quantitative and qualitative approaches to social research in human geography-an impossible mixture? *Environment and planning A*, PION LTD, v. 30, p. 261–276, 1998.
- PIVELLI, S. *Análise do potencial pedagógico de espaços não-formais de ensino para o desenvolvimento da temática da biodiversidade e sua conservação*. Tese (Doutorado) — Faculdade de Educação da USP, 2006.
- POPE, C.; MAYS, N.; POPE, C.; MAYS, N. Métodos qualitativos na pesquisa em saúde. *Pope C, Mays N, organizadores. Pesquisa qualitativa na atenção à saúde*. Porto Alegre: Editora Artmed, p. 11–9, 2005.
- PRESTRIDGE, S. The beliefs behind the teacher that influences their ict practices. *Computers & Education*, Elsevier, 2011.

- PRUDÊNCIO, A. C.; VALOIS, D. A.; LUCCA, J. E. Introdução a personalização e a localização de software. In: ROCHA IRIA SPONHOLZ, R. M. G. M. (Ed.). *Cadernos de Tradução*. [S.l.]: Editora da UFSC, 2005. Vol. 2, n. N.14, p. 211–242.
- ROCHA, L.; CASTRO, C.; MACHADO, J.; ANDRADE, R. Utilizando reconfiguração dinâmica e notificação de contextos para o desenvolvimento de software ubíquo. *XXI Simpósio Brasileiro de Engenharia de Software*, p. 219–235, 2007.
- RUTTEN, N.; JOOLINGEN, W. van; VEEN, J. van der. The learning effects of computer simulations in science education. *Computers & Education*, Elsevier, 2011.
- SAVENYE, W.; ROBINSON, R. Qualitative research issues and methods: An introduction for educational technologists. *Handbook of research for educational communications and technology*, Macmillan Library Reference USA New York, p. 1171–1195, 1996.
- SCHMIDT, D. Guest editor's introduction: Model-driven engineering. *Computer*, IEEE, v. 39, n. 2, p. 25–31, 2006.
- SCHRUM, L. Educators and the internet: A case study of professional development. *Computers & Education*, Elsevier, v. 24, n. 3, p. 221–228, 1995.
- SCHRUM, L. Technology professional development for teachers. *Educational Technology Research and Development*, Springer, v. 47, n. 4, p. 83–90, 1999.
- SEAMAN, C. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, IEEE, v. 25, n. 4, p. 557–572, 1999.
- SHEPHERD, C. Objects of interest. *Brighton East Sussex: Fastrak Consulting Limited*. Retrieved June, v. 14, p. 2005, 2000.
- SILVA, M. A. G.; BARBOSA, E. F.; MALDONADO, J. C. Model-driven development of learning objects. In: *41st Frontiers in Education Conference (FIE 2011)*. [S.l.: s.n.], 2011.
- SMITH, P.; RAGAN, T. J. *Instructional Design*. 2nd. ed. [S.l.]: Wiley, 1999.
- SNYDER, C. *Paper prototyping: The fast and easy way to design and refine user interfaces*. [S.l.]: Morgan Kaufmann Pub, 2003.
- SOLEY, R. et al. Model driven architecture. *OMG white paper*, v. 308, p. 308, 2000.
- SOUZA, M. d. F. C. d.; CASTRO-FILHO, J. A. d.; ANDRADE, R. M. d. C. Applying model-driven development for building customizable learning objects. *IEEE Multidisciplinary Engineering Education Magazine*, Vol. 6, p. 22–29, 2011.

- SOUZA, M. d. F. C. d.; CASTRO-FILHO, J. A. d.; ANDRADE, R. M. d. C. Proceedings of the 1st international symposium on open educational resources: Issues for localization and globalization. In: \_\_\_\_\_. [s.n.], 2011. cap. Guided Customization for Learning Objects, p. 105 – 125. Disponível em: <<http://educacaoaberta.org/rea/wp-content/uploads/2011/10/1symposium-proceedings.pdf>>. Acesso em: 18 jan. 2012.
- SOUZA, M. de; CASTRO-FILHO, J.; ANDRADE, R. Desenvolvendo objetos de aprendizagem utilizando modelos arquiteturais. In: *Fifht Latin American Conference on Learning Objects (LACLO 2010)*. [S.l.: s.n.], 2010.
- SOUZA, M. de; CASTRO-FILHO, J.; ANDRADE, R. Model-driven development in the production of customizable learning objects. In: IEEE. *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*. [S.l.], 2010. p. 701–702.
- SOUZA, M. de; GOMES, D.; BARROSO, G.; SOUZA, C. de; CASTRO-FILHO, J. de; PEQUENO, M.; ANDRADE, R. LOCPN: redes de petri coloridas na produção de objetos de aprendizagem. *Revista Brasileira de Informática na Educação*, v. 15, n. 3, 2007.
- STALLMAN, R. et al. The free software definition. *Free Software Free Society: Selected Essays of Richard M. Stallman*, p. 41–44, 1996.
- STERK-ELIFSON, C. Determining drug use patterns among women: The value of qualitative research methods. *NIDA Research Monograph*, National Institute on Drug Abuse, v. 157, p. 65–83, 1995.
- STRUDLER, N.; WETZEL, K. Lessons from exemplary colleges of education: Factors affecting technology integration in preservice programs. *Educational Technology Research and Development*, Springer, v. 47, n. 4, p. 63–81, 1999.
- SUBER, P. Open access overview. *Richmond: Earlham College*, 2004.
- SUBER, P. *Timeline of the open access movement*. [S.l.]: Version: <http://www.earlham.edu/~peters/fos/timeline.htm>, 2009.
- SYSTEMS, A. Adobe flash platform runtimes/statistics: Pc penetration. 2011. Disponível em: <<http://www.adobe.com/products/flashplatformruntimes/statistics.html>>. Acesso em: 18 jan. 2012.
- TAYLOR, S.; BOGDAN, R. *Introduction to qualitative research methods: A guidebook and resource*. [S.l.]: John Wiley & Sons Inc, 1998.



- TEIJLINGEN, E. van; FORREST, K. The range of qualitative research methods in family planning and reproductive health care. *Journal of Family Planning and Reproductive Health Care*, British Medical Journal Publishing Group, v. 30, n. 3, p. 171, 2004.
- TICHY, W. Hints for reviewing empirical work in software engineering. *Empirical Software Engineering*, Springer, v. 5, n. 4, p. 309–312, 2000.
- TOLVANEN, J. Implementing your own domain-specific modeling languages: Hands-on. 2011.
- WALLL, P. d.; MARCUSSO, N.; TELLES, M. *Tecnologia e Aprendizagem*. Práxis: Comunidade de Prática de Tecnologia em Educação, 2006. 45 – 47 p. (Coleção Tecnologia e Educaçã, Vol.I). Disponível em: <[http://giselebrugger.com/tutorial/Tecnologia\\_e\\_Aprend-vol1.pdf](http://giselebrugger.com/tutorial/Tecnologia_e_Aprend-vol1.pdf)>. Acesso em: 18 jan. 2012.
- WALSHAM, G. The emergence of interpretivism in is research. *Information systems research*, INSTITUTE FOR OPERATIONS RESEARCH, v. 6, n. 4, p. 376–394, 1995.
- WATSON, J.; DICKENS, A.; GILCHRIST, G. The loc tool: creating a learning object authoring tool for teachers. Association for the Advancement of Computing in Education, 2008.
- WESTBROOK, L. Qualitative research methods: A review of major stages, data analysis techniques, and quality controls. *Library & information science research*, Elsevier, v. 16, n. 3, p. 241–254, 1994.
- WILEY, D. Impediments to learning object reuse and openness as a potential solution. *Revista Brasileira de Informática na Educação*, v. 17, n. 03, p. 08, 2010.
- WILLEY, D. *Conecting learning objects to instructional theory: A definition, a methaphor anda a taxonomy*. *The Instructional Use of Learning Objects*. [S.l.]: Wiley, D.(Ed.), 2001.
- WILLIS, J.; THOMPSON, A.; SADERA, W. Research on technology and teacher education: Current status and future directions. *Educational Technology Research and Development*, Springer, v. 47, n. 4, p. 29–45, 1999.
- YUSUF, L.; CHESSEL, M.; GARDNER, T. Implement model-driven development to increase the business value of your it system. *IBM developerWorks*, 2006.
- ZATTI, V. *Autonomia e educação em Immanuel Kant e Paulo Freire*. [s.n.], 2007. Disponível em: <<http://www.pucrs.br/edipucrs/online/autonomia/autonomia-autonomia.html>>. Acesso em: 8 out. 2011.

## Apêndice A

### Mematodelo Emfatic/Eugenia do CLO Studio

```

@namespace(uri="clostudio", prefix="clostudio")
package clostudio;

@gmf.diagram(foo="bar")
class CLO {
    val Scene[*] scenes;
    val Action[*] actions;
    val Start[1] start;

    attr String[1] CLOName;
    attr String CLONick;
    attr String LOMGeneralTitle;
    attr String LOMGeneralLanguage;
    attr String LOMGeneralDescription;
    attr String LOMTechnicalSize;
    attr String LOMTechnicalLocation;
    attr String LOMEducationalInteractivityType;
    attr String LOMEducationalInteractivityLevel;
    attr String LOMEducationalTypicalAgeRange;
    attr String LOMAnnotationEntity;
    attr String LOMAnnotationDate;
    attr String LOMAnnotationDescription;
}

@gmf.node(border.width="2", label="name",
    label.icon="false", figure="rounded", ...)
class Scene extends Element {
    @gmf.compartment(foo="bar")
    val Element[*] contents;
    attr String bgImage;
    attr String bgImageDescription;
}

class Element {
    attr Boolean disable = "false";
    attr String name;
    attr int x;
}

```

```
        attr int y;
        attr int height;
        attr int width;
        attr String description;
    }

class Media extends Element {
    attr Boolean replace = "false";
    attr String sourceFile;
}

class Control extends Element {
    attr Boolean hide = "false";
}

@gmf.node(label="name", label.icon="false",
          figure="rounded", ... )
)
class Audio extends Media {
    attr String[+] event;
    attr String[+] source;
    attr String[+] action;
}

@gmf.node(label="name", label.icon="false",
          figure="rectangle", ... )
class Video extends Media {
    attr Boolean fractionate = "false";
}

@gmf.node(label="name", label.icon="false",
          figure="rectangle", ...)
class Image extends Media {
    attr Boolean hide = "false";
}

@gmf.node(label="value", label.icon="false",
          figure="rectangle", ...)
class Text extends Control {
    attr Boolean edit = "false";
    attr String value;
    attr Integer fontSize;
}

@gmf.node(label="label", figure="rounded", ...)
)
class Button extends Control {
```

```
        attr String label;
    }

@gmf.node(label="name", figure="ellipse", ...)
class Start extends Element {
}

class Action {
    ref Scene target;
}

@gmf.link(label="time", source="source",
          target="target", source.decoration="square",
          target.decoration="arrow", ...)
class TimeAction extends Action {
    ref Scene source;
    attr String[1] time;
}

@gmf.link(label="srcComponent,srcEvent,trgComponents,trgActions",
          source="source", label.pattern="Valor: {0}{1}{2}{3}",
          target="target", source.decoration="square",
          target.decoration="arrow", ...)
class MouseAction extends Action{
    ref Scene source;
    attr String[1] srcEvent;
    attr String[1] srcComponent;
    attr String[+] trgActions;
    attr String[+] trgComponents;
}

@gmf.link(source="source", target="target",
          target.decoration="arrow", ...)
class StartAction extends Action {
    ref Start source;
}
}
```

## Apêndice B

### Regras de Validação em EVL

```

context Scene{
  constraint SceneHasName {
    guard : self.isTypeOf(Scene)
    check : self.name.isDefined()
    message : 'Unnamed ' + self.eClass().name +
              ' not allowed '
  }
}
context Element{
  constraint HasNameElement {
    guard : self.isTypeOf(Button) or self.isTypeOf(Text)
           or self.isTypeOf(Video) or self.isTypeOf(Audio)
           or self.isTypeOf(Image)
    check : self.name.isDefined()
    message : 'Unnamed ' + self.eClass().name +
              ' not allowed '
  }
}
context CLO{
  constraint HasNameCLO {
    guard : self.isTypeOf(CLO)
    check : self.CLOName.isDefined()
    message : 'Unnamed ' + self.eClass().name +
              ' not allowed '
  }
}
context Scene {
  constraint NameStartsWithCapital {
    guard : self.satisfies('SceneHasName')
    check {
      var nameToLowerCase = self.name.firstToLowerCase();
      return nameToLowerCase = self.name;
    }
    message : 'Scene ' + self.name + ' should start with
              a lower-case letter '
    fix {
      title : 'Rename to ' + self.name.firstToLowerCase()
      do {

```

```
        self.name := self.name.toLowerCase();
    }
}
}
context Element {
  constraint NameWithoutBlank {
    guard : self.isTypeOf(Button) or self.isTypeOf(Text)
           or self.isTypeOf(Video) or self.isTypeOf(Audio)
           or self.isTypeOf(Image) or self.isTypeOf(Scene)
    check {
      return not(' '.isSubstringOf(self.name));
    }
    message : 'Identifier ' + self.name + ' should not
              have blank characters '
  }
}
context Element {
  constraint IdentifiersMustBeUnique {
    guard: self.name.isDefined()
    check: self.name.isUniqueIdentifier()
    message: 'Duplicate identifier: ' + self.name
  }
}
operation String isUniqueIdentifier() : Boolean {
  return Element.allInstances().select(c|c.name =
    self).size() = 1;
}
```

## Apêndice C

### Scripts de Transformação

Neste apêndice são apresentados apenas alguns trechos dos códigos dos *scripts*. Uma versão completa dos mesmos podem ser obtidos diretamente com a autora deste trabalho.

#### C.1 Geração de Templates de Implementação-CLO2Flex.m2t

```

texttransformation Transformation (in mdl:" clostudio" ) {
//Regras para escrever o mxml da aplicacao
mdl.CLO:: setStates () {
println (" public function set _ states ( state : String ) : void { " );
  self . scenes -forEach ( c : mdl . Scene ) {
    if ( c . name ! = null )
      println ( " + c . name + " . currentState = state ; " );
  }
println ( " } " );
println ( " \n " );
}
//-----
mdl.CLO:: ativarBtSelecionarversao () {
println (" public function ativar _ bt _ selecionarversao () : void { " );
self . scenes -forEach ( c : mdl . Scene ) {
if ( c . name ! = null )
  println ( " + c . name + " . selecionarversao . visible = true ; " );
}
println ( " } " );
println ( " \n " );
}
//-----
mdl.CLO:: desativarBtSelecionarversao () {
println (" public function desativar _ bt _ selecionarversao () : void { " );
self . scenes -forEach ( c : mdl . Scene ) {
if ( c . name ! = null )
  println ( " + c . name + " . selecionarversao . visible = false ; " );
}
}

```

```

println ( " } " );
println ( "\n " );
}
//-----
mdl.CLO::ativarBtCustomizar () {
println ( " public function ativar_bt_customizar () : void { " );
self.scenes-foreach ( c : mdl.Scene ) {
if ( c.name != null )
println ( " + c.name + ".bt_customizar.enabled=true ; " );
}
println ( " } " );
println ( "\n " );
}
//-----
mdl.CLO::desativarBtCustomizar () {
println ( " public function desativar_bt_customizar () : void { " );
self.scenes-foreach ( c : mdl.Scene ) {
if ( c.name != null )
println ( " + c.name + ".bt_customizar.enabled=false ; " );
}
println ( " } " );
println ( "\n " );
}
//-----
mdl.CLO::init ( CLONick : String ) {
tab(3) println ( " private function init () : void { " );
tab(4) println ( " readConfigFile ( \" users \" ); " );
tab(4) println ( " readConfigFile ( \" \" + CLONick + \" \" ); " );
tab(4) println ( " setar_eventos_mouse_action () ; " );
tab(4) println ( " setar_visibilidade_subcenas () ; " );
tab(4) println ( " setar_eventos_audio () ; " );
self.actions-foreach ( c : mdl.StartAction ) {
if ( c.target._getContainer().name == "CLO" ) {
tab(4) println ( " app.selectedChild = \" + c.target.name + \" ; " );
tab(4) println ( " TimeEvents ( \" + c.target.name + \" ); " );
}
}
tab(4) println ( " mostraVersao () ; " );
tab(3) println ( " } " );
println ( "\n " );
}
//-----
mdl.CLO::scenes_actions () {
self.actions-foreach ( c : mdl.MouseAction ) {
if ( !c.source.name.equals ( c.target.name ) ) {
println ( " public function evento_ \" + c.srcComponent + \"
(e : MouseEvent) : void { " );

```



```

    println ("app.selectedChild="+c.target.name+"");
    println ("}");
    println ("\n");
}
}
}
//-----
mdl.CLO::viewStack(){
String primeiro;
println ("mx:ViewStack id=\"app\" height=\"100%\" width=\"100%\"
        creationComplete=\"init()\" creationPolicy=\"all\"
        change=\"onViewChange(event)\" x=\"0\" y=\"0\"");
self.scenes-foreach(c:mdl.Scene c.namenu){
    println (" "+c.name.toUpperCase()+" id=\""+c.name+"\"/")
}
println ("/mx:ViewStack");
println ("\n")
}
//----- Fim da aplicacao Principal -----
//----- Regra que trata os Componentes -----
mdl.CLO::componentes(){
self.scenes-foreach(c:mdl.Scene){
    if(c.name!=null){
        file(c.name.toUpperCase()+".mxml");
        println ("?xml version=\"1.0\" encoding=\"utf-16\"?");
        println ("");
        println ("s:NavigatorContent
                xmlns:fx=\"http://ns.adobe.com/mxml/2009\"");
        println ("xmlns:s=\"library://ns.adobe.com/flex/spark\"");
        println ("xmlns:mx=\"library://ns.adobe.com/flex/mx\"");
        println ("currentStateChange=\"changeState()\"");
        println ("\n");
        println ("fx:Script source=\"FuncoesOAC.as\" /");
        println ("\n");
        println ("fx:Script");
        println ("![CDATA[");
        println ("\n");
        println ("protected function changeState():void{");
        println ("if (currentState!=\"display\"){"");
        c.contents-foreach(e:mdl.Element){
            ...
        }
    }
}
// Gera os elementos das subcenas
c.contents-foreach(q:mdl.Scene){
    if(q.name!=null){
        q.contents-foreach(d:mdl.Element){

```

```

        println("setControlsPos("+d.name+"_Canvas, "+d.name+");");
    }
}
println("}");
println("}");
println("\n");
println("]]");
println("/fx:Script");
println("\n");
println("s:states");
println("s:State name=\"display\"/");
println("s:State name=\"edit1\"/");
println("s:State name=\"edit2\"/");
println("s:State name=\"edit3\"/");
println("/s:states");
println("\n");
println("\n");
println("mx:Canvas id=\""+c.name+"\"");
//Coloca imagem de Background
if(!c.bgImage.equals("")){
    println("mx:Image creationComplete=\"setDimensions2
            (event,0,0,\"+c.height+\",\"+c.width+\",\"+c.height+\",
            "+c.width+)\",\" maintainAspectRatio=\"false\"
            source=\"@Embed(./components/\"+
            c.bgImage+".swf)\", /");
}
//Gera os elementos do Group
// Elementos do tipo Image
...
// Elementos do tipo Text
...
// Elementos do tipo Button
...
// Elementos do tipo Audio
...
// Elementos do tipo Video
...
// Gera os elementos das subcenas
c.contents--forEach(q:mdl.Scene){
    ...
}
...
println("!-- Controle de Customização --");
...
// Elementos de Customização para Subcenas
...

```

```

//----- Regra que trata a Aplicacao
mdl.CLO::aplicacao(){
file("OAC.mxml")
var CLONick : String = self.CLONick;
CLONick = CLONick.replace(" ", "");
println("xml version=\"1.0\" encoding=\"utf-16\"?");
println("");
println("s:Application
        xmlns:fx=\"http://ns.adobe.com/mxml/2009\"");
println("xmlns:s=\"library://ns.adobe.com/flex/spark\"");
println("xmlns:mx=\"library://ns.adobe.com/flex/mx\"");
println("xmlns:local=\"*\");
println("xmlns=\"*\");
println("\n");
println("fx:Script source=\"FuncoesOAC.as\" /");
println("\n");
println("fx:Declarations");
println("\n");
println("fx:XML id=\"versionsXML\" /");
println("fx:XML id=\"usersXML\" /");
println("\n");
println("/fx:Declarations");
println("\n\n");
println("fx:Script");
println("\n");
println("![CDATA[");
println("\n");
println("import mx.events.IndexChangedEvent;");
println("var timer:Timer;");
println("public var currentVideo:Object;");
println("\n");
println("\n");
println("public var oac:String = \"\"+CLONick+\"\";");
println("public var versaoatual:String = \"Original\";");
println("\n");

self.ativarBtSelecionarversao();
self.desativarBtSelecionarversao();
self.ativarBtCustomizar();
self.desativarBtCustomizar();

// Eventos de Tempo
println("public function
        onViewChange(evt:IndexChangedEvent):void{")
println("var comp:Object =
        app.getChildAt(evt.newIndex) as Object;");
println("TimeEvents(comp);");
</pre

```

```

println ("}");
self.init(CLONick);

// Gera as funções
self.funcoes();
println ("\n");
println ("]]");
println ("/fx: Script");
println ("\n");
println ("s:states");
println ("s:State name=\"display\"/");
println ("s:State name=\"edit1\"/");
println ("s:State name=\"edit2\"/");
println ("s:State name=\"edit3\"/");
println ("/s:states");
println ("\n");
self.viewStack();
println ("/s: Application");
}
mdl.CLO::funcoes(){
tab(3)println("public function
                setar_eventos_mouse_action():void{")
...
// Setar Visible nos Elementos das Subcenas
...
// Implementação dos eventos
self.scenes-forEach(c:mdl.Scene){
  c.contents-forEach(d:mdl.Audio){
    ...
  }
}
...
// Implementa os eventos
tab(3)println("public function
  change_index_events(e:IndexChangedEvent):void{");
self.scenes-forEach(c:mdl.Scene){
c.contents-forEach(d:mdl.Audio){
  var events:List;
  var actions:List;
  var sources:List;
  d.event-forEach(e){
    events.add(e);
  }
}
...
// Eventos de Tempo
tab(3)println("private function
  TimeEvents(comp:Object):void{")

```

```

tab(3) println ("// Origem da Time Action ");
tab(3) println ("// Inicia o cronômetro
    quanto ativada a Scene");
tab(3) println ("// para video se estiver sendo tocado");
tab(3) println (" if (currentVideo!=null){ ");
tab(3) println (" currentVideo.stop();");
tab(3) println ("}");
tab(3) println (" ");
self.actions-foreach(t:mdl.TimeAction){
tab(4) println (" if(comp.id==\""+t.source.name+"\"){");
tab(5) println (" timer = new Timer(\"+t.time+\"000,1)\");
tab(5) println ("// Destino da Time Action");
tab(5) println (" timer.addEventListener(TimerEvent.TIMER,
    evento_timer_\"+t.source.name+\");");
tab(5) println (" timer.start();");
tab(5) println ("}");
tab(4) println ("\n");
}
tab(3) println ("}");
// Funções que tratam da Time Actions
self.actions-foreach(t:mdl.TimeAction){
tab(4) println (" private function
    evento_timer_\"+t.source.name
    +(event:TimerEvent):void{");
tab(5) println (" app.selectedChild=\"+t.target.name+\";");
tab(5) println (" timer.stop();");
tab(4) println ("}");
}
}

mdl.CLO:: geraBloco(cena:String , controles:String){
...

println ("/mx:Canvas");
}
//-----
mdl.CLO::main(){ //execução da transformação
//Cria MXML Principal
self.aplicacao();

//Cria componentes
self.componentes();
}
}

```

## C.2 Geração de Arquivos de Simulação-CLO2FlexSim.m2t

```

texttransformation Transformation (in mdl:" clostudio" ) {
mdl.CLO::init () {
    tab(2) println("private function init():void {");
    self.actions-foreach(c:mdl.StartAction){
        if (c.target._getContainer().name=="CLO"){
            tab(3) println("app.selectedChild="+c.target.name+";");
            tab(4) println("TimeEvents("+c.target.name+");");
        }
    }
    tab(3) println("setar_eventos_mouse_action();");
    tab(3) println("setar_visibilidade_subcenas();");
    tab(2) println("}");
}
//-----
mdl.CLO::scenes_actions () {
self.actions-foreach(c:mdl.MouseAction){
if (!c.source.name.equals(c.target.name)){
println("public function evento_"+c.srcComponent
    +" (e:MouseEvent):void {");
println("app.selectedChild="+c.target.name+";");
println("}");
}
}
}
//-----
mdl.CLO::viewStack () {
String primeiro;
println("mx:ViewStack id=\"app\" height=\"100%\"
    width=\"100%\" initialize=\"init()\"
    creationPolicy=\"all\"
    change=\"onViewChange(event)\" x=\"0\" y=\"0\");
self.scenes-foreach(c:mdl.Scene c.namenu){
    println(""+c.name.firstToUpper()+" id=\""+c.name+"\"/")
}
println("/mx:ViewStack");
}
//--- Componentes
mdl.CLO::componentes () {
self.scenes-foreach(c:mdl.Scene){
if (c.name!=null){
file (c.name.firstToUpper()+".mxml");
println("?xml version=\"1.0\" encoding=\"utf-16\"?");
println("s:NavigatorContent

```

```

        xmlns:fx="http://ns.adobe.com/mxml/2009/");
println("xmlns:s="library://ns.adobe.com/flex/spark/");
println("xmlns:mx="library://ns.adobe.com/flex/mx/");
println("fx:Script");
println("![CDATA[");

println("s:Group id=\"" + c.name + "\"");

//Coloca imagem de Background
if(!c.bgImage.equals("")){
    ...
}
//Gera os elementos do Group
// Elementos do tipo Image
...
// Elementos do tipo Text
...
// Elementos do tipo Button
...
// Elementos do tipo Audio
...
// Elementos do tipo Video
...
// Gera os elementos das subcenas
...
//--- Aplicação
mdl.CLO::aplicacao(){
    file("OAC.mxml")
    var CLOName : String = self.CLOName;
    CLOName = CLOName.replace(" ", "");
    println("xml version="1.0" encoding="utf-16"?");
    println("");
    println("s:Application
        xmlns:fx="http://ns.adobe.com/mxml/2009/");
    println("xmlns:s="library://ns.adobe.com/flex/spark/");
    println("xmlns:mx="library://ns.adobe.com/flex/mx/");
    println("xmlns:local="*"");
    println("xmlns="*"");
    println("\n");
    println("fx:Script");
    println("\n");
    println("![CDATA[");
    println("import mx.events.IndexChangedEvent;");
    println("var timer:Timer;");
    println("public var oac:String = \"" + CLOName + "\"");
    println("public var versaoatual:String = \"Original\"");
    println("\n");
</pre

```

```

// Eventos de Tempo
println(" public function
    onViewChange( evt:IndexChangedEvent ): void {")
println(" var comp: Object =
    app.getChildAt( evt.newIndex ) as Object;");
println(" TimeEvents( comp );");
println(" }");
self.init();
// Gera as funções
self.funcoes();
println(" ||");
println(" /fx: Script");
self.viewStack();
println(" /s: Application");
}
mdl.CLO::funcoes(){
println(" public function
    setar_eventos_mouse_action(): void {")
self.actions-foreach( c:mdl.MouseAction){
...
}
println(" }");
...

// Setar Visible nos Elementos das Subcenas
...
// Eventos de Tempo
tab(3) println(" private function
    TimeEvents( comp: Object ): void {")
self.actions-foreach( t:mdl.TimeAction){
...
}
tab(4) println(" }");
// Funções que tratam da Time Actions
self.actions-foreach( t:mdl.TimeAction){
...
}
}
}
//-----
mdl.CLO::main(){ //execução da transformação
    self.aplicacao();
    self.componentes();
}

```



## C.3 Geração do Roteiro de Versões-CLO2Versions.m2t

```

texttransformation Transformation (in mdl:" clostudio") {
//----Regra Geral
mdl.CLO::aplicacao(){
  var CLONick : String = self.CLONick;
  CLONick = CLONick.replace(" ", "");
  file (CLONick+".xml");
  println ("?xml version = \"1.0\"?");
  println (" aplicacoes ")
  println (" aplicacao nome=\""+CLONick+"\"");
  println (" versao val=\"Original\"
descricao=\"Verso Original do OAC.\"");
  String cena="";
  self.scenes-foreach(c:mdl.Scene){
    if(c.name!=null){
      println (" cena nome=\""+c.name+"\"");
      c.contents-foreach(d:mdl.Element){
        if(d._getClass().name=="Text"){
          println (" componente ");
          println (" nome"+d.name+"/nome");
          println (" atributotext/atributo");
          println (" valor"+d._getFeature("value")+"/valor");
          println (" /componente");
        }
        if(d._getClass().name=="Image"){
          println (" componente ");
          println (" nome"+d.name+"/nome");
          println (" atributoenabled/atributo");
          println (" valortrue/valor");
          println (" /componente");
          println (" componente ");
          println (" nome"+d.name+"/nome");
          println (" atributosource/atributo");
          println (" valor"+d.name+".swf/valor");
          println (" /componente");
          println (" componente ");
          println (" nome"+d.name+"/nome");
          println (" atributovisible/atributo");
          println (" valortrue/valor");
          println (" /componente");
          println (" componente ");
          println (" nome"+d.name+"/nome");
          println (" atributoposition/atributo");
          println (" valorD:"+d.x+":"+d.y+":"+d.height+":"+
+d.width+":"+c.height+":"+c.width+"/valor");

```

```
        println(" /componente ");
    }
    if(d._getClass().name!="Scene" && d._getClass().name!="Text"
        && d._getClass().name!="Start"
        && d._getClass().name!="Image"){
        println(" componente ");
        println(" nome"+d.name+"/nome");
        println(" atributoenabled/atributo");
        println(" valortrue/valor");
        println(" /componente ");
    }
    // #####
    //Subcenas
    //#####
    if(d._getClass().name=="Scene"){
        if(d.name!=cena){
            ...
        }
    }
    println(" /cena ");
}
}
println("/versao");
println("/aplicacao");
println("/aplicacoes");
}
//-----
mdl.CLO::main(){
    self.aplicacao();
}
}
```

## C.4 Geração do Roteiro de Atividades-CLO2Fop.m2t

```

texttransformation FOPTransformation (in mdl:"clostudio"){

//----- Regra que trata as tabelas em XSL-FO
mdl.Scene::tabelasXslFo(Integer i){
  println("fo:page-sequence
          master-reference=\"A4-portrait \");
  println("fo:flow flow-name=\"xsl-region-body \");
  println("fo:block font-weight=\"bold\"
          font-family=\"any \");
  println("Cena "+i+": "+self.name);
  println("/fo:block")
  println("fo:block#160;
          fo:table table-layout=\"fixed\"
          width=\"100%\"
          display-align=\"center \");
  println("fo:table-column
          column-width=\"proportional-column-width(1)\");
  ...
  println("/fo:table-body");
  println("/fo:table");
}
...
self.contents-foreach(c:mdl.Element){
  println("fo:table-row
          border-width=\"1\" border-style=\"solid \");
  println("fo:table-cell column-number=\"1\");
  println("fo:block"+c.name+"/fo:block");
  println("/fo:table-cell");
  if(c.oclGetType()=="Image"){
    ...
  }else
  if(c.oclGetType()=="Text"){
    ...
  }
  println("fo:table-cell column-number=\"3\");
  println("fo:block"+c.description+"/fo:block");
  println("/fo:table-cell");
  println("/fo:table-row");
}
println("/fo:table-body");
println("/fo:table");
println("/fo:flow");
println("/fo:page-sequence");
}

```

```

//----- Arquivo de Saida -----
mdl.CLO::xslFoInfo(){
    file("Storyboard.fo")
    Integer i = 1;
    String primeiro;
    println("xml version=\"1.0\"?");
    println("fo:root
        xmlns:fo=\"http://www.w3.org/1999/XSL/Format\"");
    println("fo:layout-master-set");
    println("fo:simple-page-master
        master-name=\"A4-portrait\"");
    println("page-height=\"29.7cm\"
        page-width=\"21.0cm\" margin=\"2cm\"");
    println("fo:region-body /");
    println("/fo:simple-page-master");
    println("/fo:layout-master-set");
    println("fo:page-sequence
        master-reference=\"A4-portrait\"");
    println("fo:flow flow-name=\"xsl-region-body\"");
    println("fo:block text-align=\"center\"
        padding-top=\"9cm\"");
    println("font-family=\"any\" font-weight=\"bold\"");
    println(self.CLOName);
    println("/fo:block");
    println("/fo:flow");
    println("/fo:page-sequence");
    self.actions-foreach(c:mdl.StartAction){
        primeiro = c.target.name;
        c.target.tabelasXslFo(i);
        i=i+1;
    }
    self.scenes-foreach(c:mdl.Scene){
        if(c.name!=null &amp;&amp; c.name!=primeiro){
            c.tabelasXslFo(i);
            i=i+1;
        }
    }
    println("/fo:root");
}
//-----
mdl.CLO::main(){
    self.xslFoInfo();
}
}
</pre

```

## C.5 Geração de Metadados LOM-CLO2LOM.m2t

```

texttransformation Transformation (in mdl:" clostudio") {
mdl.CLO:: criarLomXml(){
  file ("LOM.xml")
  println ("?xml version =\ "1.0\ " encoding =\ "UTF-8\ "?");
  println ("lom xmlns =\ "http:// ltsc .iee e .org /xsd /LOM\ " ");
  println (" general ");
  println (" title ");
  println (" string language =\ "" + self .LOMGeneralLanguage +\ ""
    + self .LOMGeneralTitle + " string ");
  println (" /title ");
  println (" language + self .LOMGeneralLanguage + " language ");
  println (" description ");
  println (" string language =\ "" + self .LOMGeneralLanguage +\ ""
    + self .LOMGeneralDescription + "/ string ");
  println (" /description ");
  println (" /general ");
  println (" technical ");
  println (" size + self .LOMTechnicalSize + "/ size ");
  println (" location + self .LOMTechnicalLocation
    + "/ location ");
  println (" /technical ");
  println (" educational ");
  println (" interactivitytype ");
  println (" value + self .LOMEducationalInteractivityType
    + "/ value ");
  println (" /interactivitytype ");
  println (" interactivitylevel ");
  println (" value + self .LOMEducationalInteractivityLevel
    + "/ value ");
  println (" /interactivitylevel ");
  println (" typicalagerange ");
  println (" string xml : lang =\ "" + self .LOMGeneralLanguage +\ ""
    + self .LOMEducationalTypicalAgeRange + "/ string ");
  println (" /typicalagerange ");
  println (" /educational ");
  println (" annotation ");
  println (" entity ");
  println (" vcard + self .LOMAnnotationEntity + "/ vcard ");
  println (" /entity ");
  println (" date + self .LOMAnnotationDate + "/ date ");
  println (" description ");
  println (" string xml : lang =\ "" + self .LOMGeneralLanguage +\ ""
    + self .LOMAnnotationDescription + "/ string ");
  println (" /description ");
}
}

```

```
println("/annotation");
println("/lom");
}
mdl.CLO::main(){
    self.criarLomXml();
}
```

## Apêndice D

### Roteiros de Entrevista

#### D.1 Roteiro de Entrevista da Equipe de Desenvolvimento

##### Roteiro de Entrevista Semi-Estruturada

(Equipe de Desenvolvimento)

Pesquisa realizada com membros da equipe pedagógica (pedagogos e especialistas de área), design e técnica do grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (Proativa) da Universidade Federal do Ceará, utilizando a Ferramenta de Autoria, denominada de CLO Studio, para construção de OAC.

**INFORMAÇÕES GERAIS**

Nome:

**Função que exerce na equipe:**

Sexo: Masculino ( ) Feminino ( )

**Formação:**

**Tempo que desenvolve/projeta OA**

**Área:**

**Tipo de OA desenvolvido** : receptivo ( ) diretivo ( ) descoberta guiada ( )  
exploratório ( )

- Receptivo: vídeo, tutoriais e imagens
- Diretivo: exercícios práticos
- Descoberta Guiada: animação, jogos
- Exploratório: simulações

**ANÁLISE DA CONCEPÇÃO – EQUIPE PEDAGÓGICA**

1. A linguagem adotada na DSL e utilizada no CLO Studio foi de fácil compreensão?
2. A notação empregada na DSL atendeu suas necessidades para projetar o OA?
3. O CLO Studio possibilitou criar uma clara representação da estrutura do OAC?
4. Existe(m) vantagem(ns) de criar um roteiro de atividades utilizando o CLO Studio ao invés de documentos escritos? Caso haja, exemplifique



**ANÁLISE DA USABILIDADE – EQUIPE PEDAGÓGICA**

1. Você necessitou de alguma orientação para utilizar o CLO Studio ou sua utilização foi intuitiva?
2. Você necessitou de conhecimentos extras (linguagem de programação, design, etc) para utilizar o CLO Studio?
3. É fácil manipular os elementos da DSL no CLO Studio?
4. As ações de customização foram facilmente compreendidas e utilizadas?

**ANÁLISE DO DESIGN – EQUIPE DESIGN**

1. Foi possível compreender a estrutura do OAC, projetada pela equipe pedagógica, através da documentação gerada pelo CLO Studio?
2. A linguagem adotada na DSL e utilizada no CLO Studio foi de fácil compreensão?
3. A documentação gerada pelo CLO Studio possibilita compreender de modo claro o sequenciamento das ações de navegação contidas na estrutura do OAC?
4. A comunicação entre você (designer) e os membros da equipe pedagógica sofreu alguma interferência (não alterou/ melhorou/ piorou) com o uso da documentação gerada pelo CLO Studio?
5. A comunicação entre você (designer) e os membros da equipe técnica sofreu alguma interferência (não alterou/ melhorou/ piorou) com o uso da documentação gerada pelo CLO Studio?

**ANÁLISE DA PROGRAMAÇÃO – EQUIPE TÉCNICA**

1. Foi possível compreender a estrutura do OAC, projetada pela equipe pedagógica, através da documentação gerada pelo CLO Studio?
2. A linguagem adotada na DSL e utilizada no CLO Studio foi de fácil compreensão?
3. A documentação gerada pelo CLO Studio possibilita compreender de modo claro o sequenciamento das ações de navegação contidas na estrutura do OAC?
4. A comunicação entre você (programador) e os membros da equipe pedagógica sofreu alguma interferência (não alterou/ melhorou/ piorou) com o uso da documentação gerada pelo CLO Studio?
5. A comunicação entre você (programador) e os membros da equipe de design sofreu alguma interferência (não alterou/ melhorou/ piorou) com o uso da documentação gerada pelo CLO Studio?

## D.2 Roteiro de Entrevista do Professor

### Roteiro de Entrevista Semi-Estruturada

(Professor)

#### ORIENTAÇÕES PARA REALIZAÇÃO DA PESQUISA

A análise com base na perspectiva da utilização de um OA, será realizada em dois momentos.

**1º Momento:** Consiste em esclarecimentos do professor a cerca da utilização do OA monolítico, contemplando aspectos que justificam adoção do referido recurso em sala de aula.

**2º Momento:** Consiste na apresentação do OAC ao professor, pelo pesquisador e sua consequente utilização, buscando realizar uma análise sobre a percepção do professor em relação a aplicabilidade e eficácia do recurso em sala de aula.

#### INFORMAÇÕES GERAIS

**Nome:**

**Idade:**

**Sexo:** Masculino ( ) Feminino ( )

**Formação:**

**Tempo que leciona:**

**Área:**

**Nível:** Infantil ( ) Fundamental ( ) Médio ( ) Superior ( ) Profissional ( )

**Atuação:** Sala de aula ( ) LIE ( )

**Frequência com que planeja suas aulas com a utilização do OA em sala de aula:**

Às vezes ( ) Sempre ( )

## 1º Momento

Obs.: questões para serem respondidas no início da entrevista.

1. Dê exemplo de um OA que você usou em sala de aula (Descreva um pouco o OA)
2. Por que você utilizou OA em suas aulas?
3. Como você inseriu o OA em seu planejamento de aula? (Que tipo de ajuste fez no planejamento da aula?).
4. Que dificuldades você sentiu na aplicação do OA em Sala de aula? (dificuldades pessoais e dos alunos).
5. Você já deixou de utilizar um OA em sala de aula porque algum aspecto (linguagem, representação, níveis de atividades, etc) dele não atendia seu público? Qual?

## **2º Momento**

Obs.: questões para serem respondidas somente após a utilização do OAC.  
Todas as questões devem ser analisadas com base na estratégia empregada para desenvolver o OAC (Customização Guiada)

### **ANÁLISE DO CONTEÚDO**

1. Qual sua opinião sobre a possibilidade de modificar informações no conteúdo abordado pelo recurso?
2. Os ajustes permitidos no OAC são importantes para o Professor?
3. Que tipo de ajuste chamou mais sua atenção? Por que?
4. Quais as vantagens de se utilizar um OAC em relação a um OA monolítico?

### **ANÁLISE DA USABILIDADE E FUNCIONALIDADE**

1. É fácil identificar os elementos que possibilitam as adaptações?
2. Você necessitou de conhecimentos extras para modificar o recurso?
3. Você necessitou de alguma orientação para utilizar o OAC ou sua utilização foi intuitiva?
4. Os OAC são mais difíceis de manipular que os OA monolíticos? Por que?

## Apêndice E

### Diários de Campo

#### E.1 Diário de Campo para a Equipe de Desenvolvimento

##### Roteiro de Registro em Diário de Campo

##### Roteiro de Observações

**Data:**

**Membro Equipe:**

**Impressões sobre a interação da equipe pedagógica com o CLO Studio**

**Impressões sobre as ações de customização utilizadas pela equipe pedagógica**

**Impressões sobre as interações entre as equipes (pedagógica, design e técnica) através da documentação gerada a partir do CLO Studio**

**Outras informações relevantes**

## E.2 Diário de Campo para o Professor

### Roteiro de Registro em Diário de Campo

#### Roteiro de Observações

**Data:**

**Nome do(a) Professor (a):**

**Breve descrição da experiência docência do professor (a)**

**Impressões sobre a experiência do professor (a) com o OA  
monolítico**

**Impressões sobre a interação do professor (a) com o OAC**

**Ações de customização que mais chamou atenção do professor (a)**

**Outras informações relevantes**

## Apêndice F

# Roteiro de Atividades da Validação com o Professor

### Cena 1: telainicial

Descrição	Storyboard

### Background

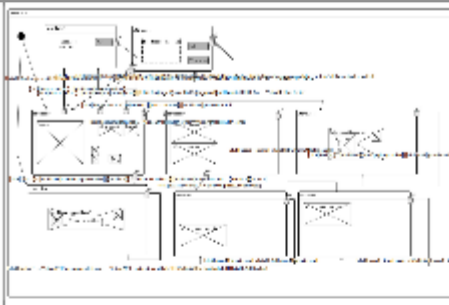
Nome	Tipo	Descrição
cenario 1	Imagem	Nessa tela o cozinheiro deverá aparecer em zoom, ou seja, ampliado e o fundo com um filtro mais escuro para que seja dado ênfase ao cozinheiro e a sua fala. A passagem dessa tela para a próxima (falacozinheiro) terá uma animação, da qual deverá haver uma mudança (em zoom) do cozinheiro para ver também o cen#ao ao fundo (cozinha). Aparecer a cozinha com alguns utensilios (1 fogão com panela em cima, uma coifa em cima do fogão, pia de cozinha, espumadeira, concha, e garfo pendurados). Colocar um balcão e o cozinheiro por trás.

### Elementos

Nome	Tipo	Descrição
cozinheiro1	Imagem	Busto do cozinheiro-Gordo, com bigode e cara de italiano. (tipo Luigi). Colocar avental e chapéu de cozinheiro.
fala_cozinheiro	Caixa de Texto	



**Cena 2: atividade1**

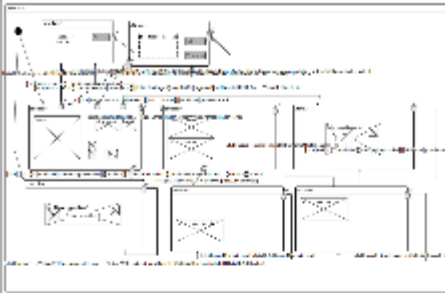
Descrição	Storyboard
	

**Elementos**

Nome	Tipo	Descrição
pizza1	Imagem	A pizza será sempre de calabresa e poderá ser dividida em no máximo 8 pedaços. Deverá aparecer em cima de um prato. No pedido os valores para denominador deverão variar de 2 a 8 e os valores para o numerador poderão variar de 1 a 7. Não poderá acontecer de o numerador ser igual ao denominador, e sempre o numerador deverá ser menor que denominador. Pra fatiar utilizará os botões fatiar mais e menos, para selecionar as fatias, o usuário terá que clicar em cada fatia da pizza.
enviarpedido1	Imagem	Será um botão com o texto: entregar pedido, esse texto também deverá aparecer em mouseover. Quando o usuário clicar nesse botão,

		aparecerá o feedback de acordo com sua resposta.
verpedido1	Imagem	Será um botão com o nome ver pedido que também aparecerá em mouseover. Quando o usuário clicar, aparecerá uma subcena com o pedido. No pedido os valores para denominador deverão variar de 2 a 8 e os valores para o numerador poderão variar de 1 a 7. Não poderá acontecer de o numerador ser igual ao denominador, e sempre o numerador deverá ser menor que denominador.
botaofatiarMais	Imagem	Nesse botão terá uma seta (para cima) e um sinal de "+" dentro da seta. Aparecerá em mouseover fatiar mais. Quando o usuário clicar nesse botão, aumentará a quantidade de fatias da pizza. Cada clique é uma fatia a mais. Esse botão poderá fatiar a pizza em no máximo 8 pedaços.
ajuda3	Imagem	Botão com imagem que simboliza ajuda. Aparecerá em mouseover o nome ajuda. Texto da ajuda: Clique no botão ver pedido para saber como será fatiada a pizza. Para fatiar a pizza, utilize os botões "fatiar +" e "fatiar -". Selecione as fatias, de acordo com o pedido, clicando na pizza. Quando finalizar clique no botão "entregar pedido".
botaofatiarMenos	Imagem	Nesse botão terá uma seta (para baixo) e um sinal de "-" dentro da seta. Colocar em mouseover "fatiar menos". Quando o usuário clicar nesse botão, diminuirá a quantidade de fatias da pizza. Cada clique é uma fatia a menos.

**Cena 3: atividade1**

Descrição	Storyboard
	

**Elementos**

Nome	Tipo	Descrição
fundopopuatividade1	Imagem	Aparecer um filtro em cima do background.
cozinheiroatividade1	Imagem	Busto do cozinheiro-Gordo, com bigode e cara de italiano. (tipo Luigi). Colocar avental e chapéu de cozinheiro.
textoatividade1	Caixa de Texto	
fechar2	Imagem	Aparecerá um botão em forma circular vermelho, com um X dentro branco. Aparecerá em mouseover a palavra: "fechar".

**Cena 4: falacozinheiro**

Descrição	Storyboard
	<p>The storyboard for 'falacozinheiro' shows a scene with a speech bubble containing the text: 'As pizzas estão prontas! Preciso de sua ajuda para levar os pedidos ao cliente. Vamos lá!'. To the left of the speech bubble is a small square icon with an 'X' inside. To the right of the speech bubble is a button labeled 'próximo' with an 'X' inside.</p>

**Background**

Nome	Tipo	Descrição
cenario2	Imagem	Aparecer a cozinha com alguns utensílios (1 fogão com panela em cima, uma coifa em cima do fogão, pia de cozinha, espumadeira, concha, e garfo pendurados). Colocar um balcão e o cozinheiro por trás.

**Elementos**

Nome	Tipo	Descrição
cozinheiro2	Imagem	Busto do cozinheiro-Gordo, com bigode e cara de italiano. (tipo Luigi). Colocar avental e chapéu de cozinheiro.
fala_cozinheiro2	Caixa de Texto	
proximo	Imagem	Imagem simboliza um botão de próximo (vai para a tela da atividade1). Aparecer o texto com o nome "próximo", em mouseover.