

PROCESSO PRODUTIVO
DE OBJETOS DE APRENDIZAGEM: PROBLEMAS E SOLUÇÕES

NÚBIA DOS SANTOS ROSA SANTANA DOS SANTOS

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY
RIBEIRO – UENF

CAMPOS DOS GOYTACAZES – RJ
JUNHO – 2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

PROCESSO PRODUTIVO
DE OBJETOS DE APRENDIZAGEM: PROBLEMAS E SOLUÇÕES

NÚBIA DOS SANTOS ROSA SANTANA DOS SANTOS

“Dissertação apresentada ao Centro de Ciência e Tecnologia da Universidade Estadual do Norte Fluminense Darcy Ribeiro, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção.”

Orientadora: CLEVI ELENA RAPKIEWICZ, D. Sc.

Co-orientador: LEANDRO KRUG WIVES, D. Sc.

CAMPOS DOS GOYTACAZES – RJ
JUNHO – 2009

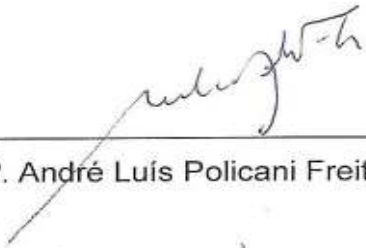
PROCESSO PRODUTIVO
DE OBJETOS DE APRENDIZAGEM: PROBLEMAS E SOLUÇÕES

NÚBIA DOS SANTOS ROSA SANTANA DOS SANTOS

“Dissertação apresentada ao Centro de Ciência e Tecnologia da Universidade Estadual do Norte Fluminense Darcy Ribeiro, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção.”

Aprovada em 05 de Junho de 2009.


Comissão Examinadora:



Prof^o. André Luís Policani Freitas, D.Sc. - UENF



Prof^o. Luiz Henrique Zeferino, D.Sc. – FAETEC



Prof^o. Leandro Krug Wives, D.Sc. – UFRGS
(co-orientador)



Prof^a. Clevi Elena Rapkiewicz, D.Sc. – UFRGS
(orientadora)

AGRADECIMENTOS

Agradeço a Deus por tudo.

A meus pais pela compreensão nos momentos de ausência.

Ao meu esposo pelo companheirismo, força e incentivo nos momentos difíceis.

A professora Clevis, minha orientadora, pela orientação e apoio durante toda a trajetória.

Ao professor Leandro da UFRGS, pelas sugestões e ajuda que enriqueceram essa dissertação.

Ao colega, Rogério, pela colaboração nesse trabalho.

Aos colegas de mestrado pela companhia e colaboração durante as disciplinas cursadas.

Ao Guilherme Ribacki pela participação no desenvolvimento do editor.

A Thiago, pela colaboração na pesquisa e na avaliação da ferramenta.

A toda equipe do projeto RIVED da UENF pela colaboração nas questões pertinentes aos Objetos de Aprendizagem.

As equipes do projeto RIVED da UFMA, UFU, UNESP, UFC e UNIFRA pela disponibilização dos Objetos de Aprendizagem e códigos-fonte para a realização da pesquisa.

A CAPES pelo suporte financeiro que permitiu a dedicação para realização desse trabalho.

A banca por participar da avaliação.

Enfim, a todos que direta ou indiretamente contribuíram para a realização desse trabalho.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE QUADROS	ix
LISTA DE ABREVIATURAS	x
RESUMO.....	xii
ABSTRACT	xiii
1. INTRODUÇÃO	1
1.1 Objetivos.....	4
1.2 Justificativa e Relevância.....	5
1.3 Estrutura da dissertação	6
2. METODOLOGIA	8
2.1 Análise do processo produtivo	8
2.2 Padronização da estrutura de informação de OA.....	8
2.3 Análise de linguagens e padrões de metadados para OA.....	9
2.4 Ferramenta para edição de elementos textuais de OA	11
2.5 Desenvolvimento do editor	14
2.6 Avaliação do protótipo da ferramenta.....	15
3. PROCESSO PRODUTIVO DE OBJETOS DE APRENDIZAGEM	16
3.1 Processo produtivo do Fábrica Virtual do RIVED	17
3.1.1 Problemas encontrados no processo.....	19
3.2 Processo produtivo da equipe RIVED-UENF	21
3.2.1 Plataformas de desenvolvimento.....	29
3.2.2 Problemas encontrados na fase de desenvolvimento dos OA.....	30
4. REPRESENTAÇÃO DA INFORMAÇÃO	34
4.1 Linguagens de marcação.....	34
4.1.1 SGML.....	35
4.1.2 HTML.....	35
4.1.3 XML.....	36
4.1.4 YAML	37
4.1.5 JSON	38
4.1.6 OGDL.....	38
4.2 Padrões de metadados para Objetos de Aprendizagem.....	40
4.2.1 LOM	40

4.2.2 SCORM	42
4.2.3 DCMI	43
5. PADRONIZAÇÃO DA ESTRUTURA DE INFORMAÇÃO	45
5.1 Análise dos elementos da estrutura de informação em diferentes equipes....	45
5.2 Especificação da estrutura de informação	51
5.3 Representação da estrutura de informação	57
6. EDITANDO A PARTE TEXTUAL DOS OBJETOS E OS METADADOS	65
6.1 Análise dos Editores	65
6.2 Casos de uso	72
6.3 Plataforma tecnológica para criação da ferramenta.....	78
6.4 Avaliação da ferramenta	81
7. CONCLUSÕES	90
7.1 Contribuições	90
7.2 Dificuldades Encontradas	94
7.3 Trabalhos Futuros	95
REFERÊNCIAS.....	98
ANEXO 1: Formulário de cadastramento do OA Balão.....	104
ANEXO 2: Exemplos de OA de equipes RIVED.....	107
ANEXO 3: XML <i>Schema</i>	111
ANEXO 4: Estrutura de informação padrão em XML	121
ANEXO 5: Cena do OA Balão na estrutura padrão	125
ANEXO 6: Questionário de avaliação de editores XML	128
ANEXO 7: Telas da ferramenta correspondentes a cada caso de uso.....	130

LISTA DE FIGURAS

Figura 1.1. Tipos de elementos em objetos de aprendizagem	3
Figura 1.2. Portal do RIVED	4
Figura 2.1. As seis fases da metodologia	8
Figura 2.2. Interface da ferramenta JUDE.....	10
Figura 2.3. Exemplo de uma descrição de Caso de Uso	14
Figura 3.1. Arcabouço de processo de <i>software</i>	17
Figura 3.2. Comparação - modelo cascata e RIVED.....	20
Figura 3.3. Etapas de criação dos OA da equipe RIVED-UENF	22
Figura 3.4. Mapa conceitual do OA Balão	23
Figura 3.5. Mapa de cenário do OA Balão	24
Figura 3.6. Roteiro do OA Balão	25
Figura 3.7. Mapa de Navegação do OA Balão	26
Figura 3.8. Modelo de Guia do Professor do OA Balão	26
Figura 3.9. Tela do OA Balão	29
Figura 3.10. XML do Objeto Balão	31
Figura 4.1. Representação de dados em blocos e linhas.....	37
Figura 4.2. Exemplo JSON.....	38
Figura 4.3. Exemplos OGDL	39
Figura 4.4. Esquema de metadados do padrão LOM.....	42
Figura 4.5. Metadados Dublin Core.....	43
Figura 5.1. OA “Balão”	48
Figura 5.2. Parte do arquivo XML do OA “Balão”	49
Figura 5.3. Diagrama de Classes da estrutura de informação proposta	52
Figura 5.4. Metadados no padrão LOM da estrutura de informação proposta	54
Figura 5.5. Metadados de conteúdos da estrutura de informação proposta	55
Figura 5.6. Parte do XML <i>Schema</i> do arquivo XML da equipe RIVED-UENF	58
Figura 5.7. Parte do arquivo XML (metadados LOM) da estrutura de informação proposta	58
Figura 5.8. Parte do arquivo XML (que trata cenas) da estrutura de informação proposta	59
Figura 5.9. Tela do OA "Balão" de acordo como a estrutura de informação padrão	60
Figura 5.10. Trecho do OA “Balão” na estrutura original.....	61

Figura 5.11. Trecho do OA “Balão” na estrutura padrão	61
Figura 5.12. OA “Balão” - Início da Estrutura de informação original do OA (a) e da Estrutura de informação padrão (b) (com metadados do padrão LOM)	62
Figura 5.13. OA “Balão” - Estrutura de informação original do OA (a) e a Estrutura de informação padrão (b) que trata cenas do OA	63
Figura 6.1. Interface no XML <i>Notepad</i>	65
Figura 6.2. Modelo <i>Text</i> e Modelo <i>Grid</i> do <i>Oxygen</i>	66
Figura 6.3. Modelo <i>Text</i> (a) e Modelo <i>Grid</i> (b) do <i>Altova</i>	66
Figura 6.4. Exemplo de uso do <i>Oxygen</i>	69
Figura 6.5. Exemplo de uso do <i>Altova</i>	70
Figura 6.6. Exemplo de uso do <i>XML Notepad</i>	71
Figura 6.7. Casos de uso do editor	73
Figura 6.8. Exemplo de um documento XML no formato DOM	79
Figura 6.9. Exemplo de código JDOM	80
Figura 6.10. Interface da Plataforma <i>Eclipse</i>	80
Figura 6.11. Interface que permite editar os metadados LOM	81
Figura 6.12. Interface da ferramenta desenvolvida	82
Figura 6.13. Interface da ferramenta	83
Figura 6.14. Interface da ferramenta- edição do cenário 9	84
Figura 6.15. Interface da ferramenta – edição do cenário 45	84
Figura 6.16. Resultado da avaliação por blocos	85
Figura 6.17. Resultado geral da avaliação do editores	86
Figura 6.18. Interface da ferramenta- exibição do arquivo XML gerado	87
Figura 7.1. Processo produtivo atual-modificado com a criação da ferramenta	93
Figura 7.2. Exemplo da Interface Padrão	96

LISTA DE QUADROS

Quadro 3.1. Descrição dos problemas dos OA	33
Quadro 4.1. Análise de linguagens	40
Quadro 5.1. Informações obtidas das equipes RIVED	46
Quadro 5.2. Comparativo das formas de representação de textos utilizadas pelas equipes RIVED Química	46
Quadro 5.3. Comparativo dos recursos dos OA.....	50
Quadro 5.4. Tipos de botões.....	56
Quadro 5.5. Tipos de Texto.....	57
Quadro 6.1. Análise dos editores	68
Quadro 6.2. Problemas comuns nos editores analisados	72
Quadro 6.3. Casos de uso.....	74
Quadro 6.4. Resolução dos problemas com a criação da ferramenta.....	86
Quadro 6.5. Casos de uso implementados	89

LISTA DE ABREVIATURAS

ADLNet - *Advanced Distributed Learning*
AICC - *Aviation Industry CBT Committee*
API - *Application Programming Interface*
CEFET - Centro Federal de Educação Tecnológica
DCMI - *Dublin Core Metadata Initiative*
DOM - *Document Object Model*
DP - *Design Pedagógico*
DTD - *Document Type Definition*
GP - Guia do Professor
HTML - *Hypertext Markup Language*
IEEE - *Institute of Electrical and Eletronic Engineer's*
IFF - Instituto Federal Fluminense
IMS - *IMS Global Learning Consortium*
ISO - *Internacional Standards Organization*
JDOM - *Java Document Object Model*
JSON - *JavaScript Object Notation*
LOM - *Learning Object Metadata*
LTSC - *Learning Technology Standards Committee*
MEC - Ministério da Educação
OA - Objeto de Aprendizagem
OGDL - *Ordered Graph Data Language*
OO - Orientação a Objetos
PCNEM - Parâmetros Curriculares Nacionais para o Ensino Médio
RDF - *Resource Description Framework*
RIVED - Rede Interativa Virtual de Educação
SAX - *Simple API for XML*
SCORM - *Sharable Content Object Reference Model*
SEED - Secretaria de Educação a Distância
SGML - *Standard Generalized Markup Language*
UENF - Universidade Estadual do Norte Fluminense
UFC - Universidade Federal do Ceará
UFMA - Universidade Federal do Maranhão

UFU - Universidade Federal de Uberlândia

UML - *Unified Modeling Language*

UNESP - Universidade Estadual de São Paulo

UNIFRA - Centro Universitário Franciscano

XML - *Extensible Markup Language*

XSD - *XML Schema Definition*

XSLT - *eXtensible Stylesheet Language for Transformation*

YAML - *YAML Ain't a Markup Language*

W3C - *World Wide Web Consortium*

RESUMO

O avanço no desenvolvimento de Tecnologias da Informação e Comunicação (TICs) tem permitido o uso de diferentes recursos no processo de ensino e aprendizagem, entre eles os Objetos de Aprendizagem (OA). No Brasil a produção de OA é feita em diferentes contextos, entre eles, as equipes financiadas pelo Ministério da Educação (MEC) no âmbito do projeto RIVED/Fábrica Virtual. Essas equipes têm diferentes composições segundo a instituição de ensino superior a qual estão associadas, mas em qualquer caso há uma equipe de domínio e uma equipe técnica. Essa dissertação analisou o caso das equipes de Química, entre as quais se encontra uma equipe da UENF. Verificou-se a falta de padronização na estrutura de informação bem como variações no processo produtivo. Um dos gargalos identificados foi o fato da edição dos metadados e dos dados textuais ser feita pela equipe técnica, enquanto o entendimento do que deve constar nesses elementos encontra-se com a equipe de domínio. Essa característica demanda várias interações entre as equipes que pode ser diminuída no caso da tarefa de manipulação dos metadados e dados textuais ser feita pela equipe de domínio. As várias interações geram retrabalho. Para que seja possível essa transferência de tarefa para a equipe de domínio, torna-se necessário o uso de uma ferramenta (editor) para os metadados e dados textuais. A análise de editores genéricos existentes bem como o uso dos mesmos pelo usuário mostrou-se difícil o processo de edição, demandando o desenvolvimento de um editor específico. A padronização da estrutura de informação e representação da mesma em XML *Schema* e o desenvolvimento de um protótipo de editor para manipulação do arquivo XML padronizado permitiu diminuir a quantidade de interações entre as equipes técnica e de domínio. Um experimento inicial sugere que essas duas providências (existência de um esquema padronizado e uso do editor específico pela equipe de domínio) permitem diminuir o tempo de desenvolvimento de OA.

Palavras-chave: objetos de aprendizagem, metadados, processo produtivo, XML.

ABSTRACT

Advances in the development of Information and Communication Technologies (ICT) allow the use of different resources in the teaching and learning process. Learning Objects are one of these resources, and the production of Learning Objects is made on different contexts. In Brazil, for example, production teams funded by the Ministry of Education (i.e., MEC), through the RIVED or the Virtual Factory projects, usually have different compositions of people from different contexts, according to the institution in which they are developed. On the majority of these compositions, we can identify pedagogical and technical teams. This master thesis examined some learning object production teams from the Chemistry domain, in which the UENF team is found. As a result, it was found a lack of standardization in the information structure of the objects as well as variations in the production process conducted by them. One important bottleneck identified was the metadata editing step being performed by the technical team, while the understanding of these elements must be met with the pedagogical team. This involves many iterations and interactions among teams, and if the pedagogical team could handle this task alone, rework would be minimized. To transfer this task to the pedagogical team, a tool (editor) to enable direct manipulation of metadata is necessary, along with proper consistency check. The analysis of available different editors showed that they are too generic and tend to difficult the editing process to people without technical skills, indicating that a specific editor should be proposed and built. The standardization of the information structure, its representation using XML Schema, and the development of a specific editor, all these tasks diminished the amount of interaction needed between the technical and the pedagogical team. An initial experiment suggests that these measures or precautions do reduce the development time of learning objects.

Keywords: learning objects, metadata, production process, XML.

1. INTRODUÇÃO

Vivemos na Era da informação caracterizada por Castells (1999, p.426) como constituída de redes de produção, poder e experiência, que constroem a cultura da virtualidade nos fluxos globais, os quais transcendem o tempo e o espaço. Nessa sociedade as Tecnologias de Informação e Comunicação (TICs) são fundamentais e abrangem todos os segmentos. Segundo Takahashi (2000, p.156) as TICs são tecnologias utilizadas para tratamento, organização e disseminação de informações. No contexto da educação as TICs modificaram substancialmente a base da educação uma vez que impõem aos professores outra postura que não mais o de simples assimilador/repassador de conteúdos e informações (PRETTO e SERPA, 2001).

Vários tipos de recursos de TIC podem ser utilizados como auxílio ao processo de ensino e aprendizagem, como o computador, vídeos, televisão, rádio e etc. Entre as TICs destacam-se também os objetos de aprendizagem (OA) os quais podem ser definidos:

[...] como qualquer recurso, suplementar ao processo de aprendizagem, que pode ser reusado para apoiar a aprendizagem. O termo objeto educacional (*learning object*) geralmente aplica-se a materiais educacionais projetados e construídos em pequenos conjuntos com vistas a maximizar as situações de aprendizagem onde o recurso pode ser utilizado. A idéia básica é a de que os objetos sejam como blocos com os quais será construído o contexto da aprendizagem [...] (TAROUCO, FABRE e TAMUSIUNAS, 2003, p. 2)

De acordo com Santanchè e Lago (2007), existem características essenciais para definir um OA, sendo elas: reusabilidade, estar no formato digital, ser mais do que código executável (não se restringindo a programas de computador), possuir aderência a padrões abertos e serem voltados à aprendizagem. Os OA normalmente são compostos por recursos da própria plataforma de desenvolvimento e das demais mídias já existentes, tais como: som, imagem, vídeos e textos. O grande problema está em como estruturar esses elementos dentro de um objeto.

De acordo com Longmire (2000) *apud* Souza, Andrade e Castro Filho (2008), há diversos fatores que favorecem o uso dos OA na área educacional como: a flexibilidade, a facilidade para atualização, a customização, a interoperabilidade, o aumento do valor de um conhecimento e a facilidade de indexação e procura. No entanto, os benefícios provenientes do uso de OA tornam-se mais visíveis, quando Wiley (2002) os define como pequenos componentes instrucionais que podem ser

reutilizados inúmeras vezes, em diferentes contextos de aprendizagem. Ou seja, uma das características que mais se destaca nos objetos é a reusabilidade.

A reusabilidade de OA pode ser entendida de várias formas. Uma delas refere-se ao reuso por usuários, ou seja, o destaque é o fato de poder ser utilizado em vários contextos de aprendizagem, conforme apontado na definição de Tarouco, Fabre e Tamusiunas (2003). A reutilização pelos usuários é que dá suporte a flexibilidade destacada por Longmire (2000). A dificuldade maior, no entanto, não tem ocorrido no plano educacional e sim no contexto do processo de desenvolvimento, conforme apontado por Wiley (2008).

A reusabilidade, em última instância, diz respeito a possibilidade de diminuir o tempo de desenvolvimento de artefatos de *software*. Segundo Pressman (2006) um componente de *software* deve ser projetado e implementado de modo que possa ser reusado em muitos programas diferentes. Segundo o mesmo autor, o reuso poupa tempo e esforço, e tem sido proclamado como um importante benefício do uso de tecnologias orientadas a objetos.

Considerando que os OA são constituídos de elementos diversos (som, imagem, vídeos, animações e textos), conforme ilustrado na Figura 1.1, a reusabilidade pode ocorrer através de qualquer tipo de componente. No contexto dessa dissertação, o que é tratado é o reuso das partes textuais¹ de OA. Os textos são informações como explicação de conteúdo, fala de personagens, palavras existentes nos cenários e informações para ajudar o usuário a entender o OA.

¹ Parte textual diz respeito a todo e qualquer elemento presente no objeto em forma de texto, como explicações sobre um OA ou recursos de um OA, fala de personagens, etc.

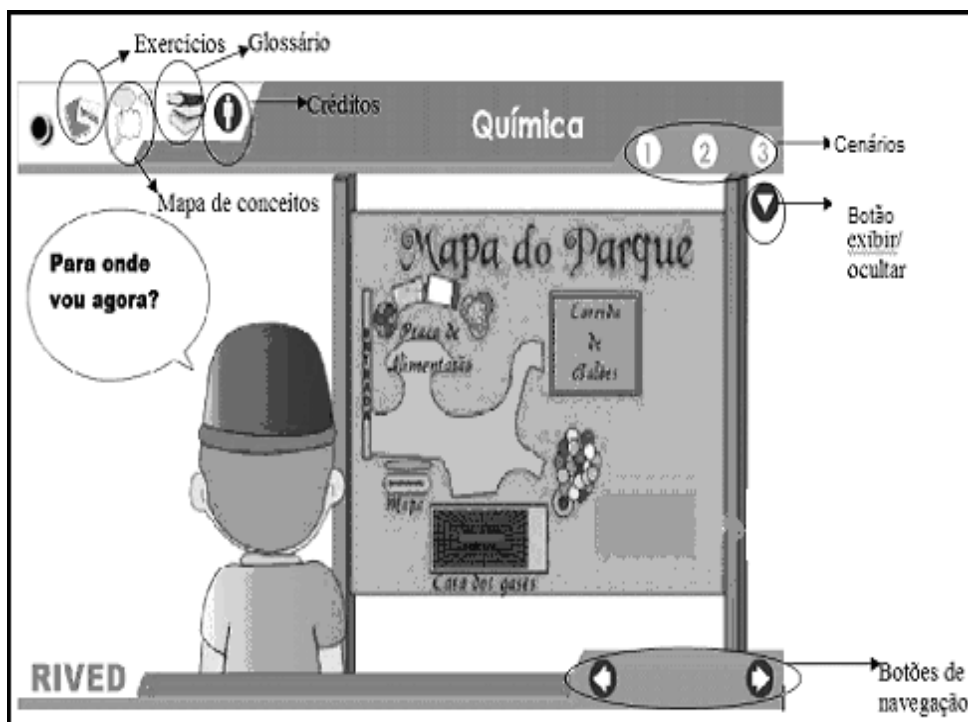


Figura 1.1. Tipos de elementos em objetos de aprendizagem

Fonte: Equipe RIVED-UENF

Considerando a impossibilidade de tratar da reusabilidade de objetos em contexto amplo no âmbito de uma dissertação de mestrado, o foco desta foi restrito a Rede Interativa Virtual de Educação (RIVED). Trata-se um programa da Secretaria de Educação a Distância (SEED) do Ministério da Educação (MEC) que tem por objetivo a produção de conteúdos pedagógicos digitais, na forma de objetos de aprendizagem (OA). Tais conteúdos têm como foco estimular o raciocínio e o pensamento crítico dos estudantes, associando o potencial da informática às novas abordagens pedagógicas. Além de promover a produção de conteúdos digitais e a publicação na *Web* para acesso gratuito, o RIVED realiza capacitações sobre a metodologia para produzir e utilizar os objetos de aprendizagem nas instituições de ensino superior e na rede pública de ensino (RIVED, 2007).

Como expansão do projeto RIVED, o Ministério da Educação (MEC) criou a Fábrica Virtual, um projeto que tem como propósito intensificar e transferir o processo de desenvolvimento e produção de recursos educacionais digitais (na forma de OA) da SEED para as Instituições de Ensino Superior e inserir novas abordagens pedagógicas que utilizem a informática nas licenciaturas das universidades por meio da promoção de um trabalho colaborativo e interdisciplinar dentro da academia (RIVED, 2007). A interdisciplinaridade ocorre através da composição das equipes, de diferentes áreas, uma técnica (responsável pelos

aspectos computacionais) e outra de domínio ou pedagógica (responsável pelos aspectos pedagógicos e pelo conteúdo). As equipes que compõem o RIVED estão localizadas nos estados do Rio de Janeiro, São Paulo, Rio Grande do Sul, Ceará, Alagoas, Paraíba, Maranhão e Goiás. São 27 equipes de 17 Instituições de ensino superior e CEFETs (Centros Federais de Educação Tecnológica) (SEEDNET, 2007).

A Figura 1.2 mostra o Portal do Projeto RIVED/Fábrica Virtual, no qual é possível ter acesso a vários Objetos de Aprendizagem de diversos domínios.



Figura 1.2. Portal do RIVED

Fonte: <http://rived.proinfo.mec.gov.br>

Conforme já destacado, o foco desta dissertação está no aspecto da reusabilidade da parte textual de OA no contexto do projeto Fábrica Virtual. Nesse contexto, a seguir são delimitados os objetivos dessa dissertação bem como a hipótese na qual foi desenvolvido o processo de pesquisa.

1.1 Objetivos

O objetivo principal dessa dissertação é melhorar o processo produtivo de OA permitindo que a manipulação da parte textual dos OA seja feita pela equipe de domínio e não pela equipe técnica, evitando o retrabalho e facilitando, dessa forma, o reuso. Esse objetivo foi delineado considerando-se a hipótese de que a manipulação das partes textuais dos objetos pela equipe de domínio diminui o tempo

de desenvolvimento através da eliminação tanto do retrabalho quanto do número de versões necessárias para o objeto ser concluído.

A partir do objetivo principal pode-se destacar objetivos específicos, tais como:

- Facilitar o processo de edição da parte textual, permitindo maior eficácia e controle sobre o conteúdo do OA.
- Evitar que erros na parte textual sejam cometidos pela equipe técnica, quando o texto do Roteiro é transferido para a parte textual do OA.
- Permitir que a equipe pedagógica tenha facilidade na edição da parte textual do OA, evitando a ocorrência de erros na exibição do conteúdo do OA.
- Fornecer ao usuário condições de alteração da parte textual dos OA sem a necessidade do conhecimento dos aspectos técnicos de linguagens utilizadas para representação.
- Possibilitar que a parte textual dos OA siga um padrão, de modo a facilitar o uso e a edição de conteúdos dos OA.
- Permitir que a ordem de apresentação do conteúdo dos OA seja alterada, sem causar danos na apresentação do conteúdo como um todo.
- Permitir que novos elementos² sejam inseridos, assim como a exclusão dos existentes, de forma que isso não prejudique a funcionalidade do arquivo e a perda de conteúdo.
- Possibilitar a inserção de caracteres especiais e formatação no conteúdo a ser exibido, de modo prático e sem a necessidade de alteração no código-fonte do OA.
- Obter uma estrutura de informação que seja padronizada e ao mesmo tempo flexível. A estrutura de informação de um OA contém elementos principais que o descrevem como um todo, e outros elementos que descrevem os recursos existentes em cada uma de suas cenas (SANTOS, RAPKIEWICZ e WIVES, 2008a).

1.2 Justificativa e Relevância

Como visto anteriormente, esta dissertação está inserida no contexto do RIVED, um programa do MEC. O RIVED é uma política pública que promove a produção e publicação de conteúdos pedagógicos digitais na Internet para acesso

² O foco dessa dissertação inclui somente os elementos textuais. Futuramente outros tipos de elementos podem ser trabalhados.

gratuito. Ainda realiza capacitações da metodologia para produzir e utilizar os OA nas instituições de ensino superior e na rede pública de ensino. É um programa de nível federal, composto de várias equipes. Uma lista das equipes, das Universidades e das áreas de domínio que compõem o Fábrica Virtual do RIVED pode ser encontrada no endereço: http://www.rived.mec.gov.br/Portal_objeto_lis.php.

Por se tratar de uma política pública, que consome recursos públicos é importante que o processo produtivo dos conteúdos pedagógicos digitais seja eficaz. De acordo com a ABNT NBR ISO 9000 (2000), processo é o conjunto de atividades inter-relacionadas ou interativas que transformam insumos (entradas) em produtos (saídas). No entanto, o processo produtivo do RIVED/Fábrica Virtual e da equipe RIVED-UENF apresenta problemas, os quais são apresentados no capítulo 3. Como o projeto utiliza recursos públicos é importante que a produção seja maximizada de modo a evitar o atraso e o desperdício de recursos. Uma das maneiras de aumentar a produtividade é diminuir o tempo de desenvolvimento através da diminuição do retrabalho.

Conforme ficará claro na sequência de capítulos, uma forma de otimização do processo produtivo, proposta desta dissertação, foi o *design* de um *software*. Este *software* é uma ferramenta de edição da parte textual dos OA.

1.3 Estrutura da dissertação

Essa dissertação está estruturada em sete capítulos. O presente capítulo, de introdução, apresentou o contexto em que está inserida a dissertação, delimitando os objetivos, a justificativa e a relevância desse trabalho.

A metodologia aplicada visando a obtenção dos objetivos propostos é apresentada no capítulo 2. A metodologia está dividida em seis fases. A primeira delas consistiu na análise do processo produtivo de objetos de aprendizagem no contexto do projeto Fábrica Virtual do RIVED. A fase seguinte foi a coleta de informações para propor uma estrutura de informação padrão para os OA. A terceira etapa referiu-se ao estudo de possíveis linguagens para representar a estrutura de informação proposta, bem como padrões de metadados³ para Objetos de Aprendizagem. A quarta foi a elicitação de requisitos para a criação de um editor específico para manipular a parte textual dos OA, sendo esse associado à

³ Metadados são informações estruturadas que descrevem, explanam, localizam, ou facilitam recuperar, usar ou gerenciar um recurso de informação (HODGE, 2001).

linguagem escolhida para representação da estrutura, já que a análise de editores genéricos mostrou que os mesmos não eram adequados ao processo proposto. A criação da ferramenta concluiu o processo de busca pela solução do problema abordado na dissertação.

O terceiro capítulo apresenta o processo produtivo de OA do projeto RIVED-Fábrica Virtual, assim como o processo produtivo da equipe RIVED-UENF, apresentando algumas diferenças quanto ao modelo proposto pelo RIVED. Ainda são apresentados os problemas encontrados em ambos os processos.

No quarto capítulo são apresentadas algumas linguagens de marcação, como SGML (*Standard Generalized Markup Language*), HTML (*Hypertext Markup Language*), XML (*Extensible Markup Language*), YAML (*YAML Ain't a Markup Language*), JSON (*JavaScript Object Notation*), OGDL (*Ordered Graph Data Language*) e suas respectivas características. São apresentados também, alguns padrões de metadados para OA, como o LOM (*Learning Object Metadata*) e SCORM (*Sharable Content Object Reference Model*).

O capítulo 5 apresenta a análise de OA de diferentes equipes RIVED de Química, a padronização da estrutura de informação e a representação dessa estrutura padrão.

O sexto capítulo apresenta a análise de editores genéricos, as plataformas e modelos necessários para o desenvolvimento de uma ferramenta específica e a avaliação da ferramenta desenvolvida. O sétimo capítulo apresenta as conclusões, destacando as contribuições dessa dissertação, apontando as dificuldades encontradas e as possibilidades para trabalhos futuros.

Por último, são listadas as referências utilizadas durante a pesquisa de todo o conteúdo abordado nesse trabalho, bem como os respectivos anexos.

2. METODOLOGIA

A metodologia aplicada para obtenção dos resultados desejados está dividida em fases como exposto na Figura 2.1.



Figura 2.1. As seis fases da metodologia

A primeira fase consistiu na análise do processo produtivo de OA no contexto do projeto Fábrica Virtual do RIVED. A segunda fase foi a coleta de informações com o objetivo de analisar os diferentes recursos existentes em cada OA para propor uma estrutura de informação padrão. A terceira fase referiu-se ao estudo de possíveis linguagens para representar a estrutura de informação proposta e padrões de metadados para OA. A quarta fase foi a elicitação de requisitos para a criação de um editor específico para manipular a parte textual dos OA, já que a análise de editores mostrou que os mesmos não eram adequados ao processo proposto. A quinta fase foi o desenvolvimento da ferramenta utilizando as tecnologias analisadas. A sexta fase foi a avaliação do protótipo da ferramenta com o objetivo de verificar a resolução dos problemas. As fases são detalhadas a seguir.

2.1 Análise do processo produtivo

Essa fase teve como objetivo analisar o processo produtivo de OA. O objetivo da análise foi conhecer o processo produtivo do RIVED/Fábrica Virtual, a fim de verificar se existiam problemas e de que forma o processo poderia ser melhorado. Depois foi analisado o processo produtivo da equipe RIVED-UENF com o objetivo de identificar possíveis problemas e buscar a melhoria do processo.

2.2 Padronização da estrutura de informação de OA

A segunda fase foi a de levantamento de requisitos para a criação de uma estrutura padrão. O levantamento de requisitos foi realizado através da análise dos diferentes recursos existentes nos OA. Essa análise foi feita com OA das equipes

RIVED. Como o projeto RIVED possui muitas equipes divididas em diversas áreas de domínio, a análise de todos os OA seria inviável. Por isso, a amostra a ser pesquisada se restringiu aos OA das equipes RIVED do domínio da Química, devido ao fato de os OA da equipe RIVED-UENF pertencerem a esta área.

A coleta de informações analisou quais recursos (imagens, tipos de botões, tipos de textos, som, vídeo e etc.) os OA possuem e de que forma eles são representados. Além disso, foi verificado se os OA são padronizados. Após a coleta, o próximo passo foi a análise e validação das informações dos OA das diferentes equipes através de um *checklist*. Esta análise dos OA das equipes RIVED teve como propósito verificar se os OA já possuem uma estrutura de informação e como esta é representada.

2.3 Análise de linguagens e padrões de metadados para OA

Após a coleta e análise das informações, foi feita a análise de linguagens de marcação existentes e de padrões para metadados de OA. O objetivo foi analisar linguagens como YAML, JSON, OGD, SGML e HTML, além da XML, para verificar qual delas é mais adequada para representar a estrutura de informação. A estrutura de informação é composta de metadados que descrevem um OA de acordo com o padrão LOM e metadados que descrevem recursos que estão efetivamente em um OA (parte textual). Além da linguagem, a análise estendeu-se ao uso de um padrão de metadados para OA.

Posteriormente, foi elaborado um modelo que representa a estrutura padrão. Para tanto, foi utilizado um Diagrama de Classes UML (*Unified Modeling Language*). De acordo com Larman (2004), um Diagrama de Classes ilustra as especificações para classes de *software* e interfaces (por exemplo, interfaces Java) em uma aplicação. Ainda, segundo o mesmo autor, a informação típica inclui:

- classes, associações e atributos;
- interfaces, com suas operações e constantes;
- métodos;
- informação de tipo de atributo;
- navegabilidade;
- dependências.

O importante é saber que o primeiro passo consiste em identificar as classes que fazem parte da solução do *software*.

Para a montagem do Diagrama de Classes é necessário utilizar um padrão de descrição de OA, o qual sirva também como referência para a coleta das informações.

A ferramenta utilizada para elaboração do diagrama de classes foi o JUDE (*Java and UML Developers' Environment*). JUDE (Figura 2.2) é uma ferramenta gratuita de modelagem para sistemas que suporta UML. A ferramenta engloba diagramas da UML, tais como diagramas de caso de uso, diagrama de atividades e diagrama de classes.

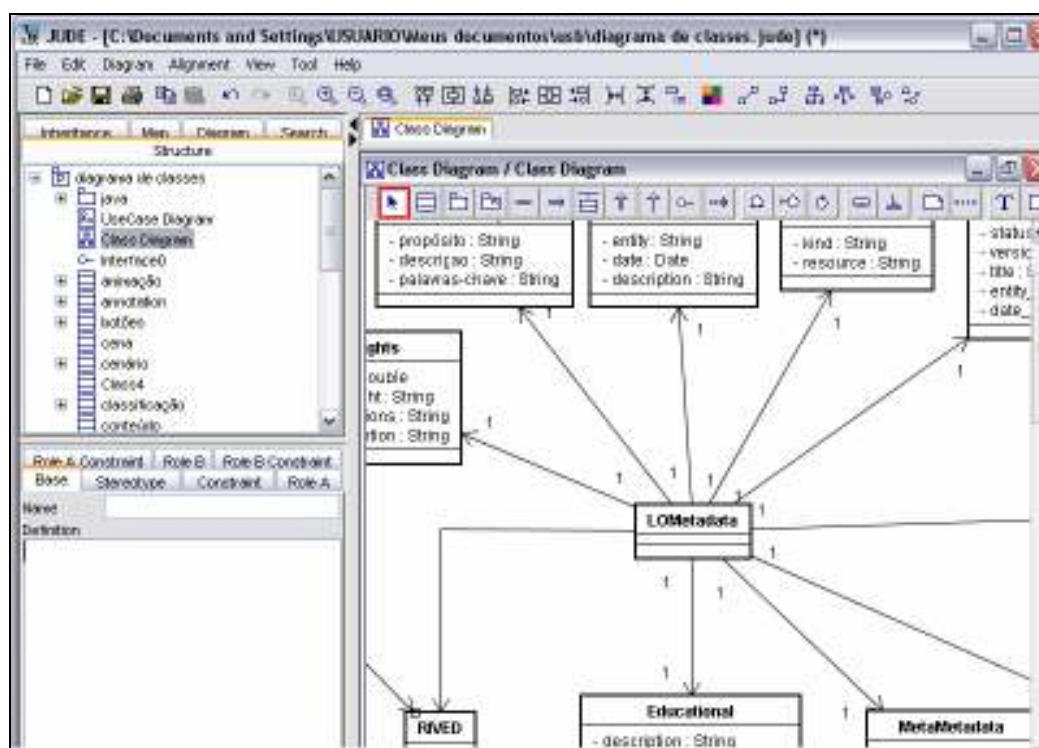


Figura 2.2. Interface da ferramenta JUDE

Uma das características importantes dos OA é que são elementos instrucionais baseados no paradigma da orientação a objetos, o que valoriza a criação de componentes que podem ser reutilizáveis em múltiplos contextos (WILEY, 2000). Os OA, como recursos educacionais, devem ser estruturados e categorizados (metadados) de modo a alcançar acessibilidade, interoperabilidade, durabilidade e reusabilidade. A utilização de padrões na descrição de metadados dos OA facilita a busca, avaliação, aquisição e o uso desses OA por estudantes ou instrutores ou mesmo por processos de *software* (SCHEER e GAMA, 2004).

Padronizados, os metadados também facilitam o compartilhamento e o intercâmbio de OA, permitindo o desenvolvimento de catálogos e inventários,

levando em conta a diversidade de contextos nas quais os objetos e seus metadados podem ser reutilizados (IEEE/LTSC 2003 *apud* SCHEER e GAMA 2004).

O modelo que representa a estrutura de informação dos OA (Capítulo 5) foi validado através de uma Prova de Conceito. De acordo com Wysocki (2006), a Prova de Conceito pode corresponder a:

- criação de um protótipo;
- um estudo de viabilidade;
- casos de uso;
- *storyboarding*;
- qualquer atividade que demonstre o valor de negócio.

Enfim, a Prova de Conceito serve para averiguar se a estrutura padrão desenvolvida consegue representar um OA, mostrando se os requisitos foram atendidos. Para realizar a Prova de Conceito foi utilizado o OA Balão, que possuía uma estrutura de informação antiga representada em XML. Na realização da Prova de Conceito, todos os recursos do OA foram descritos na estrutura padrão proposta. Além de possibilitar a descrição de todos os recursos, assim como de todas as informações da estrutura antiga, a estrutura padrão permitiu a descrição de outras informações necessárias ao OA e que eram descritas em outros locais (documentos) anteriormente.

2.4 Ferramenta para edição de elementos textuais de OA

Essa fase compreendeu, inicialmente, a análise de editores existentes, a fim de verificar se eles atendiam aos requisitos para solução dos problemas apresentados quanto à edição da estrutura de informação dos OA da equipe RIVED-UENF. Essa análise incluiu editores gratuitos e proprietários. Dentre os aspectos analisados, buscou-se verificar se os editores permitiam a edição das *tags* e seus conteúdos, se possuíam mais de um modelo de edição, como modelo de texto ou de árvore, se possibilitavam a validação do documento XML, dentre outros.

A análise foi feita pela equipe de domínio, sendo esta aqui representada pelo termo genérico “usuário”. O usuário utilizou editores genéricos (XML *Notepad*, *Altova*, *Oxygen*) para editar a estrutura de informação de um OA de acordo com o

modelo proposto. Após utilizar os editores, o usuário relatou as ações realizadas, apresentando os pontos positivos e negativos de cada editor.

Para análise dos editores, também foi utilizado um questionário, o qual foi respondido pelo usuário. O questionário está organizado em três partes: A, B e C, sendo que os critérios da parte A e B se baseiam na ABNT NBR ISO/IEC 9126 (2003).

A NBR ISO/IEC 9126, sob o título geral "Engenharia de *software* - Qualidade do produto", contém quatro partes: Modelo de qualidade, Métricas externas, Métricas internas e Métricas de qualidade em uso. Dessas partes, métricas externas e métricas internas foram utilizadas. O modelo de métricas externas e internas define o modelo de qualidade externa e interna. Ele categoriza os atributos de qualidade de *software* em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade), as quais são, por sua vez, subdivididas em subcaracterísticas. As subcaracterísticas podem ser medidas por meio de métricas externas e internas (ABNT NBR ISO 9126, 2003).

A parte A do questionário refere-se aos critérios de funcionalidade baseados na ABNT NBR ISO/IEC 9126, a parte B refere-se aos critérios de usabilidade da ABNT NBR ISO/IEC 9126 e a parte C aos critérios baseados nos requisitos para um editor de OA.

Das seis características, foram utilizadas somente funcionalidade e usabilidade. As outras não foram utilizadas porque não eram relevantes para a avaliação

tendo em vista o objetivo proposto para o desenvolvimento do *software*. O questionário utilizou a escala *Likert*. A escala *Likert*, proposta por Rensis Likert em 1932, é uma escala onde os respondentes são solicitados não só a concordarem ou discordarem das afirmações, mas também a informarem qual o seu grau de concordância/ discordância. A cada célula de resposta é atribuído um número que reflete a direção da atitude do respondente em relação a cada afirmação. A pontuação total da atitude de cada respondente é dada pelo somatório das pontuações obtidas para cada afirmação (MATTAR, 1997).

Para fazer a análise, atribui-se a cada afirmação um escore numérico que vai de -2 a +2 ou de 1 a 5. A análise pode ser feita item por item (análise de perfil) ou então pode-se calcular um escore total (somatório) para cada entrevistado somando-

se os itens. A abordagem baseada no somatório é a usada com maior frequência (MALHOTRA, 2006).

Os blocos A e B tiveram peso 3, enquanto o bloco C teve peso 4 por entender que tais questões eram fundamentais para verificar o diferencial da ferramenta em relação aos editores analisados. Para avaliar cada questão foi utilizada uma escala de 5 pontos com as opções: S (Sim), P+ (Parcialmente com poucas restrições), P (Parcialmente), P- (Parcialmente com muitas restrições) e N (Não). Os valores referentes a cada opção foram distribuídos assim:

- S vale 1,0;
- P+ vale 0,75;
- P vale 0,5;
- P- vale 0,25;
- N vale 0,0.

No bloco C os valores das questões C1, C2, C3, C5 e C6 tiveram os valores invertidos porque essas questões são inversas às outras, os símbolos (valores) foram trocados para dar um valor correto, sem erros. Por exemplo, se a questão C2 (O *software* permite renomear o conteúdo das *tags*) recebe S (1.0) então o valor é trocado para N (0), uma vez que a permissão para renomear o conteúdo das *tags* é um dos problemas mencionados nessa dissertação e, portanto, é um ponto negativo.

Considerando-se que nenhum dos editores analisados demonstrou ser adequado, devido ao fato de serem genéricos, permitirem a edição de conteúdo das *tags*, não utilizarem padrões e etc., procedeu-se o desenvolvimento de um editor específico. Os requisitos do mesmo foram identificados na fase de análise do processo produtivo de OA e também a partir da análise dos editores genéricos, considerando-se que aspectos não contemplados por esses devem estar presentes no editor desenvolvido. O registro dos requisitos foi feito através de Casos de Uso. Os Casos de Uso servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa de como o sistema é utilizado. Um cenário de caso de uso é uma sequência de passos que descreve uma interação entre o usuário e o sistema (FOWLER, 2005), conforme exemplificado na Figura 2.3.

<p>Compra de um Produto</p> <p>Nível do Objetivo: Nível do Mar</p> <p>Cenário Principal de Sucesso:</p> <ol style="list-style-type: none"> 1. O cliente <u>navega pelo catálogo e seleciona itens para comprar</u> 2. O cliente vai para o caixa 3. O cliente preenche o formulário da remessa (endereço de entrega; opção de entrega imediata ou em três dias) 4. O sistema apresenta a informação completa do faturamento, incluindo a remessa 5. O cliente preenche a informação de cartão de crédito 6. O sistema autoriza a compra 7. O sistema confirma imediatamente a venda 8. O sistema envia uma confirmação para o cliente por e-mail <p>Extensões:</p> <p>3a. Cliente regular</p> <ol style="list-style-type: none"> .1: O sistema mostra a informação atual da remessa, a informação de preço e a informação de cobrança .2: O cliente pode aceitar ou escrever por cima desses padrões, retornando ao CPS, no passo 6 <p>6a. O sistema falha na autorização da compra a crédito</p> <ol style="list-style-type: none"> .1: O cliente pode inserir novamente a informação do cartão de crédito ou cancelar

Figura 2.3. Exemplo de uma descrição de Caso de Uso

Fonte: Fowler (2005)

O caso de uso da Figura 2.3 possui um nome (Compra de um Produto), uma descrição detalhada de todas as ações e também apresenta Extensões para explicar as ações dos atores em casos específicos ou que requerem mais detalhamento.

2.5 Desenvolvimento do editor

Essa fase diz respeito ao desenvolvimento⁴ da ferramenta para manipulação das partes textuais do objeto e a sua implementação.

A linguagem de programação utilizada foi Java. Como características que motivam o uso da linguagem Java, destacam-se algumas das citadas por Deitel e Deitel (2003):

- é completamente orientada a objetos (OO) com forte suporte para técnicas adequadas de Engenharia de *Software*;
- pode ser baixada gratuitamente;
- é uma linguagem independente de plataforma, ou seja, seu desenvolvimento permite que seus programas sejam executados em diferentes

⁴ Guilherme Ribacki, estudante de Ciência da Computação da UFRGS, colaborou na finalização da ferramenta.

máquinas, mesmo com diferentes configurações de *hardware* ou sistemas operacionais;

- suporta vários recursos de alto desempenho.

A IDE (*Integrated Development Environment*) utilizada para desenvolvimento da ferramenta foi *Eclipse*. Por IDE entende-se um ambiente voltado para o desenvolvimento de *software*. *Eclipse* é uma das ferramentas mais utilizadas por programadores Java. O *Eclipse* permite o desenvolvimento baseado em *plugins*, o que possibilita atender diferentes necessidades dos programadores.

2.6 Avaliação do protótipo da ferramenta

A última fase foi a avaliação da ferramenta, que serviu para verificar a funcionalidade e aspectos de usabilidade por parte do usuário. Para isso foram utilizados os mesmos instrumentos para avaliação dos editores genéricos, isto é, relato do usuário sobre a utilização da ferramenta e um questionário.

3. PROCESSO PRODUTIVO DE OBJETOS DE APRENDIZAGEM

No contexto digital, os OA são *softwares*, uma vez que abrangem instruções executadas em computadores, seja através de navegadores ou através de arquivos executáveis, conforme são desenvolvidos (SANTOS, RAPKIEWICZ e WIVES, 2008a). Desse modo, devem seguir o processo de desenvolvimento de *software*.

De acordo com Pressman (2006), o arcabouço de processo genérico (Figura 3.1) aplicável à maioria dos projetos de *software* é composto de cinco atividades, sendo elas:

a) Comunicação: envolve alta comunicação e colaboração com o cliente, e abrange o levantamento de requisitos e atividades relacionadas;

b) Planejamento: descreve as tarefas técnicas a serem conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos de trabalho a serem produzidos, assim como o cronograma de trabalho;

c) Modelagem: inclui a criação de modelos que permitem ao desenvolvedor e ao cliente entender os requisitos de *software* e o projeto que irá satisfazer esses requisitos;

d) Construção: combina geração de código e os testes necessários para revelar erros no código;

e) Implantação: o *software* é entregue ao cliente, que avalia o produto e fornece um *feedback* com base nessa avaliação.

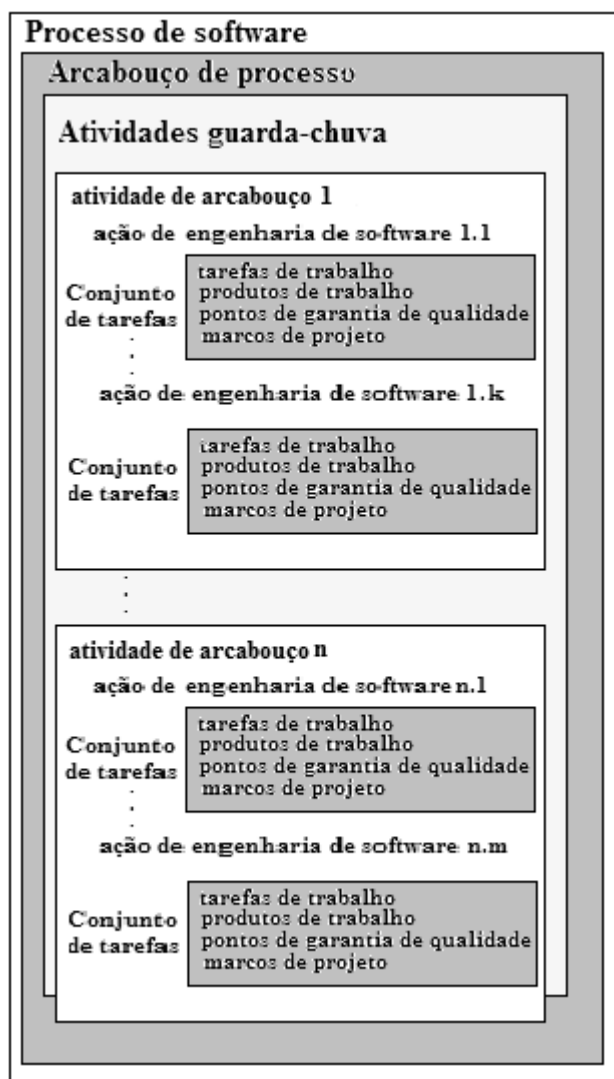


Figura 3.1. Arcabouço de processo de *software*

Fonte: Pressman (2006, p.18)

Como os OA são *softwares*, o processo de desenvolvimento deve estar de acordo com o arcabouço proposto por Pressman. Para isso, a seção 3.1 analisa o processo produtivo de OA do Fábrica Virtual em geral e alguns problemas relacionados. A seção 3.2 analisa o processo produtivo de OA da equipe RIVED-UENF em particular e os problemas encontrados no mesmo.

3.1 Processo produtivo do Fábrica Virtual do RIVED

O processo produtivo de OA do Fábrica Virtual possui várias etapas, desde a concepção até a validação do produto. Algumas variações podem ser encontradas dependendo da equipe. Para o desenvolvimento de um OA, enquanto produto, três fases são requeridas: planejamento (modelagem), desenvolvimento e validação. A fase de validação, que objetiva verificar se um OA atende aos requisitos, não é

formalmente sugerida no modelo RIVED. No processo produtivo é necessária a interação entre duas equipes, a de domínio (ou pedagógica) e a técnica, conforme destacado no primeiro capítulo.

As equipes são compostas por especialistas em conteúdo, *designer* instrucional, pedagogo, programadores (técnicos), ilustrador e *designers* de multimídia. Os especialistas das áreas disciplinares fazem o mapeamento de conteúdos e selecionam o tópico do novo módulo. Módulos são compostos por atividades que podem ter diversos tipos de recursos, possuindo OA de vários tipos (PRATA, 2006). O *designer* instrucional interage com os especialistas para criar as atividades. A equipe pedagógica define os objetivos educacionais. Os programadores programam o OA. O ilustrador e o *designer* auxiliam na criação das imagens e animações.

A equipe de domínio é responsável pela criação de documentos, correspondentes a diferentes fases do processo, como o *Design Pedagógico*, Roteiro e Guia do Professor.

O *Design* descreve as estratégias pedagógicas e atividades, como definição de conteúdo/tema/conceito, definição de mídias a serem utilizadas, o público a que se destina, contextualização, quantidade de informação, competências a serem desenvolvidas e outros (PRATA, 2006). O Roteiro descreve as especificações dos OA (PRATA, 2006), apresentando passo a passo como um OA deve ser produzido, utilizando imagens e/ou figuras para especificar como as interfaces devem ser. O Roteiro é a descrição detalhada de todas as telas que irão compor o OA (ABREU *et al.*, 2006). O Guia do Professor é um material que enriquece o repertório do professor com dicas e sugestões importantes para a condução das atividades envolvendo o OA (PRATA, 2006).

As fases de desenvolvimento dos OA do projeto RIVED, descritas por Nascimento e Morgado (2003), são:

- Fase 1 - A equipe pedagógica define os objetivos educacionais e elaboram as respectivas estratégias educacionais. O *designer* instrucional interage com especialistas a fim de traçar a sequência instrucional e o nível cognitivo requerido nas atividades. Os objetivos e as atividades são descritos em um documento denominado *Design Pedagógico* (DP). O *Design Pedagógico* é utilizado para mostrar o objetivo educacional do objeto, o tema central e as estratégias de aprendizagem para o aluno.

- Fase 2 – O documento *Design* Pedagógico é submetido às outras equipes de produção do OA para obter críticas e *feedback*. A revisão do DP consiste em comentários sobre: a) o *design* do programa e a abordagem pedagógica utilizada; b) questões relacionadas ao uso apropriado da tecnologia no objeto; c) sugestões para diferentes atividades ou mídias.

- Fase 3 – Os especialistas de conteúdo revisam o *design* original após receber *feedback* e passam a desenvolver as especificações para cada OA, na forma de *scripts* e roteiros de tela, para o grupo de técnicos desenvolver os produtos desejados. A elaboração do Roteiro ajuda a visualizar o produto final e pode reduzir frustrações e o tempo de produção.

- Fase 4 – O grupo de técnicos produz os OA. Durante essa fase, os especialistas de conteúdo, o *designer* instrucional, e os técnicos interagem bastante para evitar erros.

- Fase 5 – Os especialistas em conteúdo criam um guia do professor (GP) para cada OA. O GP pode conter exemplos e curiosidades para auxiliar o professor durante a utilização do OA (ABREU *et al.*, 2006).

- Fase 6 – Os OA são organizados em módulos (Biologia, Química, Física, Biologia, Matemática, Geografia, etc.) e publicados na *Web*.

3.1.1 Problemas encontrados no processo

Cordeiro *et al.* (2007) destacam que o modelo do processo de desenvolvimento de objetos pelo RIVED segue as mesmas diretrizes empregadas pelo modelo em cascata (ciclo de vida clássico), modelo que segue uma abordagem sistemática e linear. O modelo Cascata é um modelo projetado para ser aplicado no desenvolvimento do *software*. Conforme visto em Bassani *et al.* (2006), no modelo em cascata as etapas são realizadas de forma linear, e nesse formato sempre que houver alguma modificação, todo o processo deve ser revisto.

A Figura 3.2 mostra o modelo produtivo de OA do Fábrica Virtual sendo comparado ao modelo cascata (Cordeiro *et al.* 2007). No modelo em cascata só é possível ir para a fase seguinte após concluir a fase anterior. O modelo do RIVED mostra um fluxo em sentido único, ou seja, só é possível ir para a fase seguinte após concluir a anterior.

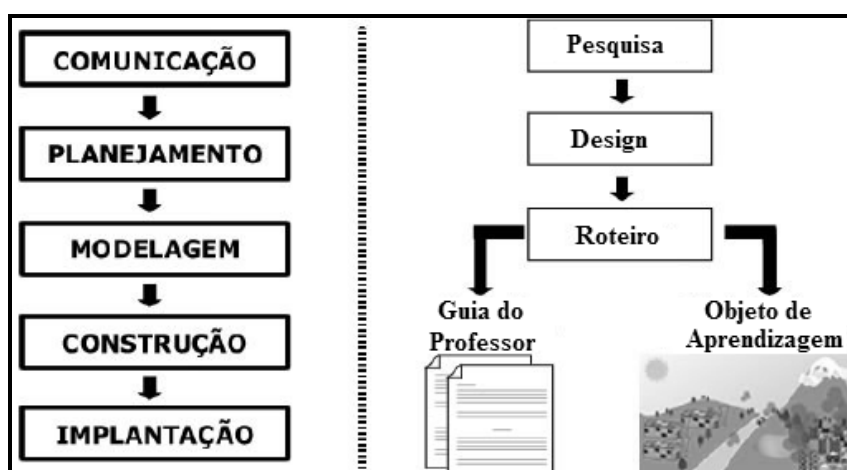


Figura 3.2. Comparação - modelo cascata e RIVED

Fonte: Cordeiro *et al.* (2007, p.4)

Os problemas relacionados ao modelo cascata, segundo Pressman (2006) são:

- Projetos reais raramente seguem o fluxo que o modelo propõe;
- É difícil para o cliente estabelecer todos os requisitos explicitamente;
- Uma versão executável do programa só ficará disponível ao final do intervalo de tempo do projeto.

Os problemas característicos desse ciclo de vida também são observados na produção de um OA. Dessa forma, esse modelo possui a desvantagem de exigir que ambas as equipes possuam bom entendimento dos requisitos, tendo estes bem definidos, pois este modelo não prevê “ciclos de realimentação” (CORDEIRO *et al.*, 2007). Como o modelo não prevê ciclos torna-se difícil adequar novas adaptações no desenvolvimento dos OA.

O modelo atual do RIVED não apresenta, explicitamente, momentos de refinamento, onde o grupo multidisciplinar (equipe de domínio e equipe técnica) possa discutir e interagir de modo a aprimorar as especificações dos OA (AMARAL *et al.*, 2006). Isso se torna um problema, uma vez que a discussão e a interação entre as equipes são fundamentais para esclarecer as dúvidas.

O Roteiro também apresenta problemas. É um documento estritamente linear que dificulta a visão do todo por parte da equipe pedagógica. Também dificulta ao entendimento da estrutura de navegação de um OA por parte da equipe técnica (CORDEIRO *et al.*, 2007).

Outro problema citado por Amaral *et al.* (2006) é que o modelo também não possui uma fase de validação junto aos usuários finais (estudantes e professores) antes da publicação de um OA para testar sua efetividade. A fase de validação é fundamental para o processo produtivo de OA, uma vez que é nela que será verificado se o objeto realmente auxilia no processo de aprendizagem.

Conforme foi apontado, seguir o modelo cascata sugerido pelo RIVED é problemático, razão pela qual algumas equipes introduziram mudanças no processo. É o caso da equipe RIVED-UENF, conforme apresentado na próxima seção.

3.2 Processo produtivo da equipe RIVED-UENF

A equipe RIVED-UENF vem buscando solucionar alguns problemas encontrados no processo do Fábrica Virtual, inclusive acrescentando etapas ao processo. No entanto, nenhuma mudança foi feita em relação ao tratamento das informações dos recursos que compõem um OA.

A equipe RIVED-UENF desenvolve OA na área de domínio de Química. A composição da equipe segue a orientação do MEC (RIVED, 2008), com interações entre uma equipe técnica e uma equipe de domínio.

A criação dos OA utiliza três etapas, sendo elas: Planejamento ou Modelagem, Desenvolvimento e Validação. A Figura 3.3 ilustra as etapas do processo produtivo de OA da equipe RIVED-UENF através de um diagrama de atividades. A notação de um diagrama de atividades é similar a de um fluxograma e é usada para representar o que acontece quando o sistema executa suas funções (PRESSMAN, 2006). Pode-se verificar que existem vários atores (equipes), um em cada raia (coluna) do diagrama de atividades.

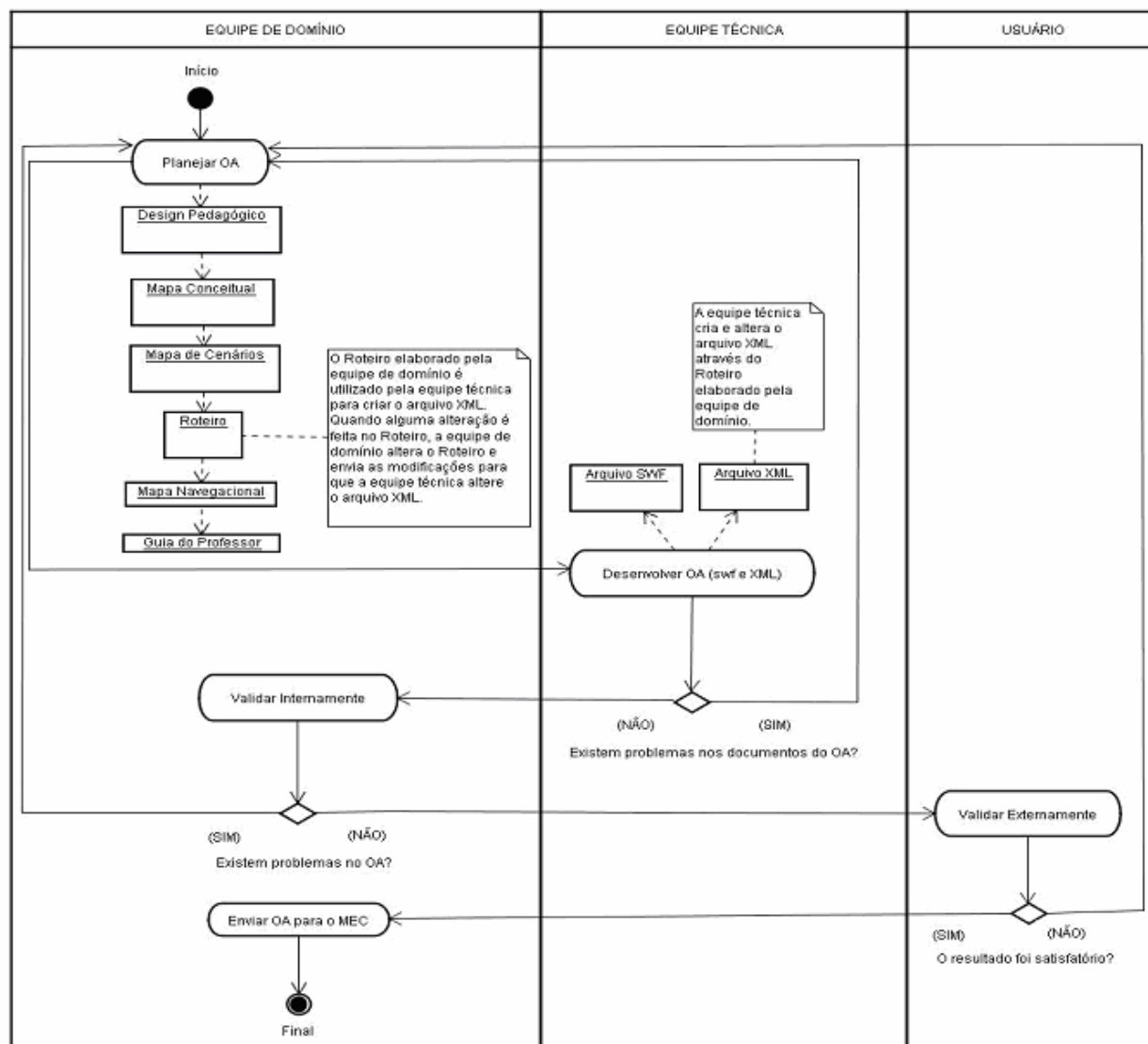


Figura 3.3. Etapas de criação dos OA da equipe RIVED-UENF

Fonte: Santos, Rapkiewicz e Wives (2008a, p.5)

Na etapa de Planejamento ou Modelagem são definidos os requisitos de um OA pela equipe pedagógica. Os documentos produzidos são os sugeridos pelo RIVED, *i.e.*, o *Design* Pedagógico, Roteiro e o Guia do Professor, conforme já citado. Além desses documentos, a equipe RIVED-UENF elabora outros três documentos: Mapa de Conceito, Mapa de Cenário e Mapa Navegacional.

O Mapa de Conceito e o Mapa de Cenário tornam mais claro o *Design* Pedagógico, auxiliam na abordagem dos conteúdos propostos e na contextualização. O termo “Mapa de Conceito” surgiu da metodologia de Mapas Conceituais de Novak (NOVAK e CANÃS, 2006), que procura representar como o conhecimento é armazenado na estrutura cognitiva de um aluno.

O Mapa de Conceito (vide Figura 3.4) tem por finalidade levantar os conteúdos (conceitos) que devem ser abrangidos por um OA (CORDEIRO *et al.*, 2007). Novak (1984) *apud* Lima (2004 p.136) refere-se a duas características básicas do mapa conceitual: "os conceitos representados de forma hierárquica, com o conceito mais geral no início do mapa e depois os mais específicos, arranjados hierarquicamente."

De acordo com Tavares (2006), um mapa conceitual pode ser entendido como uma teia de conceitos, onde fica nítida a inter-relação entre eles. O uso de mapa conceitual pode explicitar os diversos conceitos utilizados em uma animação e, desse modo, o mapa conceitual pode atuar como estruturador global do conhecimento que está sendo focalizado (TAVARES, 2006).

A equipe RIVED-UENF utiliza esse recurso para apresentar graficamente os conceitos que o OA deve abranger. O Mapa Conceitual apresentado na Figura 3.4 mostra os conceitos abrangidos pelo OA Balão, como por exemplo, temperatura, calor, pressão, gás, etc.

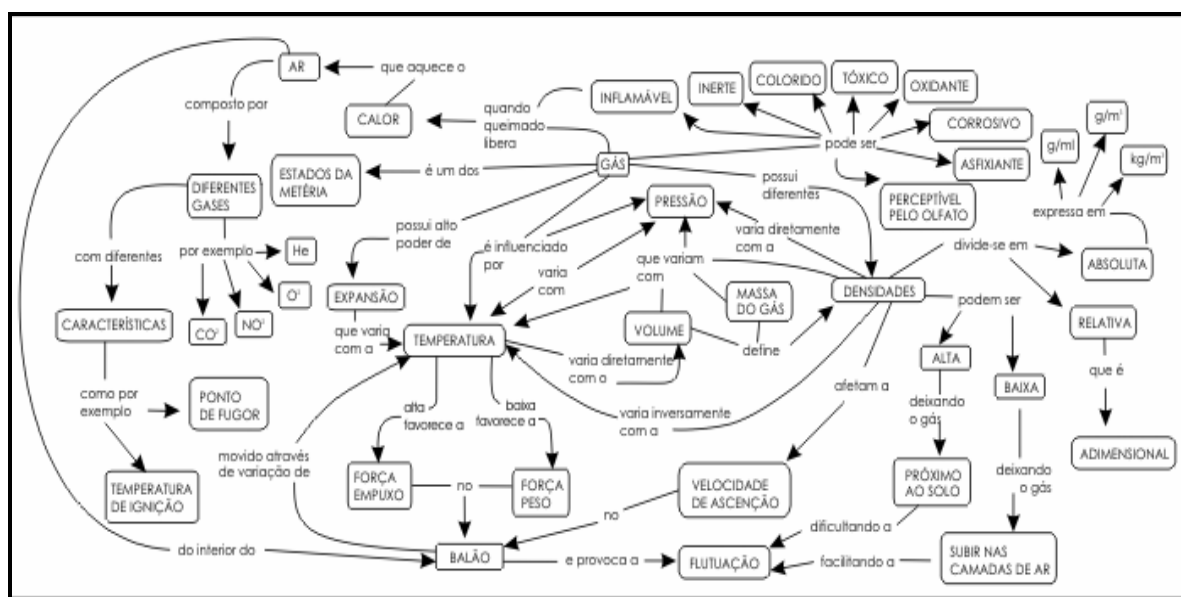


Figura 3.4. Mapa conceitual do OA Balão

Fonte: Equipe RIVED-UENF

O segundo mapa, Mapa de Cenário, é a delimitação dos cenários que serão utilizados para desenvolver os conceitos propostos por um Mapa de Conceitos (CORDEIRO *et al.*, 2007). Cada nó do mapa indica um cenário (contexto) no qual um determinado conteúdo (mostrado nas arestas de ligação entre os nós) pode ser

trabalhado com o aluno (CANELA *et al.*, 2007). O mapa de cenários apresentado na Figura 3.5 mostra os cenários, subcenários e os conteúdos que são abordados no OA Balão desenvolvido pela equipe RIVED-UENF. Por exemplo, a Figura 3.5 mostra que o *cenário* utilizado para contextualizar o estudo da influência da temperatura na densidade é uma barraca de cachorro quente e de sorvete.

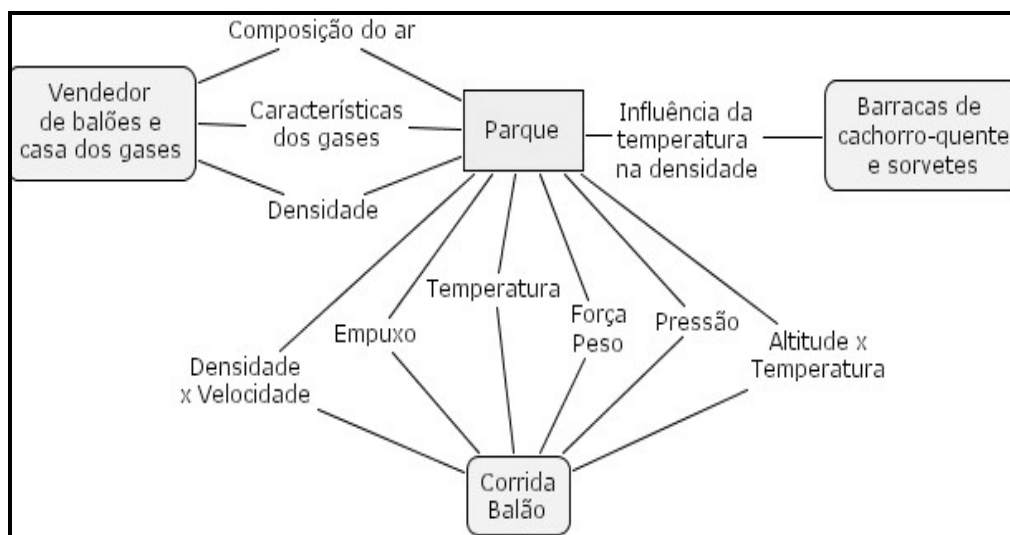


Figura 3.5. Mapa de cenário do OA Balão

Fonte: Equipe RIVED-UENF

O Roteiro (Figura 3.6) contém os textos, imagens e a descrição de como a animação deve ser executada, assim como a navegação que o usuário irá seguir. Nele estão todas as informações sobre os recursos que um OA deve ter e como esses recursos devem funcionar.

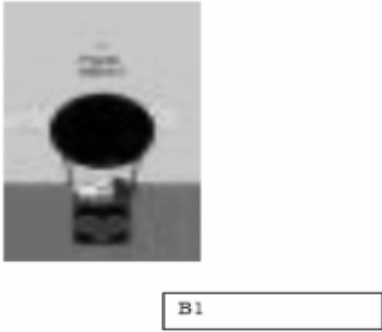
Roteiro	
Título da animação: Utilidade da Camada de Ozônio Autor: Gláucia Gonzaga	Tela 1
Texto: T1 - Vamos nos divertir explorando a atmosfera de balão? B1 - Próximo B2 - Créditos	Imagem 
Explicação sobre a ação: O usuário: <ul style="list-style-type: none"> > Lê as informações. > Clica em seguir. Descrição dos objetos: <ul style="list-style-type: none"> > B2 - vai pra tela 20 > Aparece T1 e o balão ao fundo com o fogareiro aceso. Quando o usuário clica em Próximo, o balão levanta vôo e muda para a tela 2. 	
Título da animação: Utilidade da Camada de Ozônio Autor: Gláucia Gonzaga	Tela 2

Figura 3.6. Roteiro do OA Balão

Fonte: Equipe RIVED-UENF

O Mapa Navegacional é utilizado para tornar o Roteiro mais claro. Serve como um auxílio para o desenvolvedor, fornecendo uma visão geral da navegabilidade entre as várias partes que compõem um OA, *i. e.*, mostra tanto a navegação entre cenários, quanto a navegação interna a esse. Ainda, auxilia a equipe de domínio a verificar se há algum erro, como a ausência de alguma parte em um OA. O Mapa Navegacional apresentado na Figura 3.7 mostra como deve ser a navegabilidade entre os cenários de um OA, como por exemplo, é possível acessar a Casa dos Gases a partir da Barraca de Sorvete e vice-versa.

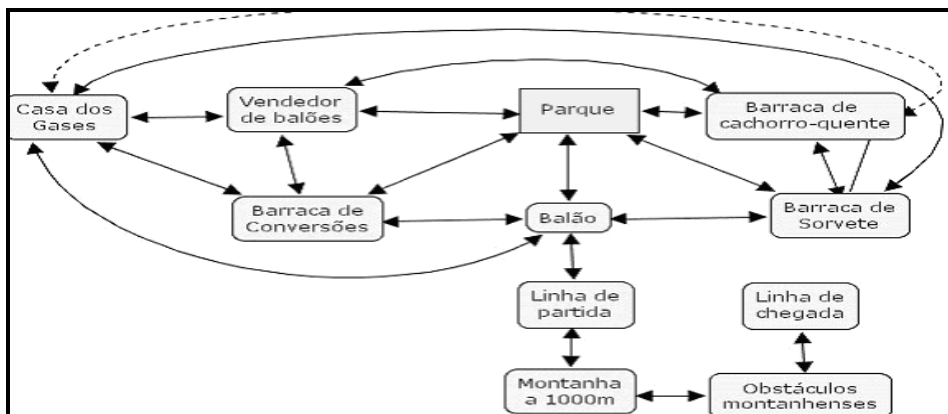


Figura 3.7. Mapa de Navegação do OA Balão

Fonte: Equipe RIVED-UENF

Na etapa de Desenvolvimento, a equipe técnica começa a produção de um OA a partir do Roteiro, dos mapas de cenário e navegacional, de acordo com as especificações feitas pela equipe de domínio. Em paralelo, a equipe de domínio constrói o GP. O GP (Figura 3.8) possui informações sobre o funcionamento de um OA e fornece sugestões aos professores de como utilizá-lo com os alunos, incluindo o tempo estimado de uso de um OA em sala de aula.

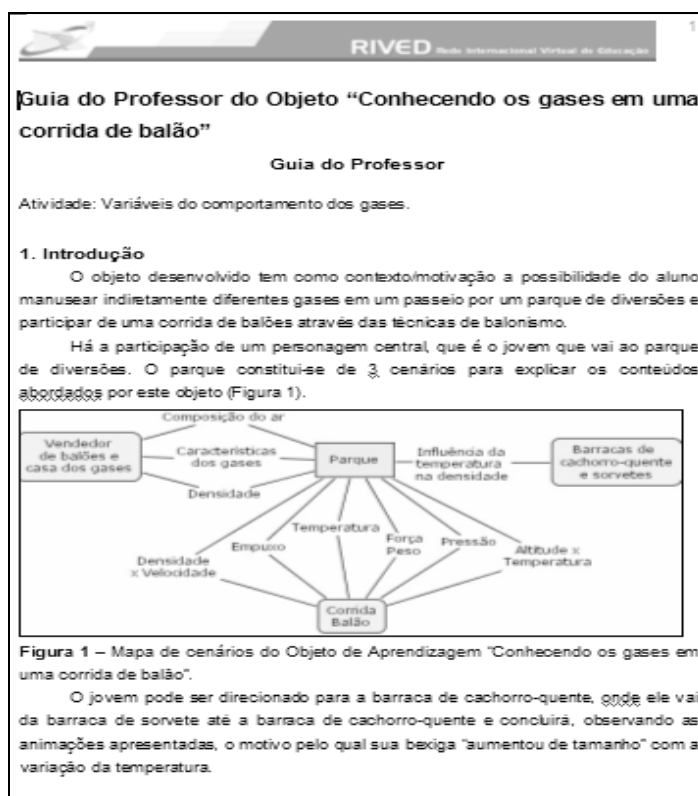


Figura 3.8. Modelo de Guia do Professor do OA Balão

Fonte: Equipe RIVED-UENF

A etapa de Validação visa verificar se o OA construído está de acordo com o planejamento para que possíveis correções sejam feitas. Apesar de o modelo RIVED não incluir uma etapa de validação no processo produtivo, equipes do Fábrica Virtual, como a equipe RIVED-UFU (SILVA e FERNANDES, 2007) e a equipe RIVED-UENF, inseriram a etapa de validação em seus processos. Essa fase objetiva verificar se o objetivo proposto foi atingido. Também são analisados os aspectos de navegabilidade, interatividade e usabilidade. De posse dos comentários e observações, a equipe técnica faz as alterações necessárias nos OA. A equipe de domínio também atua na etapa de validação, verificando se um OA atende aos requisitos e aos documentos elaborados. Essa validação não ocorre apenas ao fim do desenvolvimento de um OA, mas sim durante todo o processo, sendo mais visível na etapa de validação em si. A etapa de validação é composta de validação interna (pela equipe de domínio e técnica) e validação externa (pelos usuários).

A fase de validação utiliza a metodologia proposta por Costa *et al.* (2006), dividida em quatro categorias: pedagógica, conteúdo, interface e usabilidade. Das quatro categorias, as três primeiras são realizadas através da validação interna. Na última categoria é realizada a validação externa. A primeira categoria (pedagógica) tem como objetivo avaliar o potencial de um OA como ferramenta de ensino-aprendizagem e é subdividida em: critérios pedagógicos, critérios pedagógicos baseados nas propostas dos **Parâmetros Curriculares Nacionais para o Ensino Médio** (PCNEM) relativas à área de domínio de um OA (Química) e critérios relativos a conteúdos específicos. A segunda categoria (conteúdo) foca na qualidade e na fidedignidade das informações contidas nos OA. A terceira categoria (interface) analisa se as formas de comunicação do OA com o usuário (cores, textos, imagens, ícones, interação etc.) são eficientes e adequadas. E a quarta categoria (usabilidade) verifica se um OA é fácil de ser manipulado e navegado. Na quarta categoria, a validação é feita com alunos de ensino médio, através da aplicação de questionários, após oficinas nas quais os OA são manipulados.

Analisando a Figura 3.3, pode-se notar as mudanças que foram feitas em relação ao processo proposto no âmbito do projeto Fábrica Virtual, como a inserção de uma 3ª fase, a de validação, e alguns produtos complementares, como o mapa de conceitos, mapa de cenários e o mapa navegacional. Além disso, a equipe RIVED-UENF realiza reuniões das equipes para esclarecer as dúvidas e dar

sugestões sobre os OA, utilizando, inclusive, tais recursos gráficos para ter uma visão nítida de um OA como um todo.

Outra mudança ocorreu na composição da equipe técnica, com a inserção de um *Designer* Gráfico e de uma Avaliadora de Interfaces e Supervisão, especialista em Comunicação. A função do *Designer* é auxiliar a elaboração gráfica de uma interface atraente e intuitiva. O primeiro passo é criar a interface padrão, pela qual todos os objetos se baseiam. Com base nos roteiros e descrições de cenários e personagens fornecidos pela equipe pedagógica, o *Designer* faz os desenhos das telas e personagens, utilizando *softwares* de desenho e de edição de imagem.

É necessária a interação entre *Designer*, equipe pedagógica e a equipe técnica. A equipe pedagógica deve estar de acordo com as imagens produzidas, avaliando se as mesmas condizem com a concepção inicial de um OA, assim como a viabilidade da utilização das imagens para elaboração das animações feitas pela equipe técnica.

Já a avaliadora de Interface e Supervisão analisa a interface no intuito de verificar problemas de usabilidade tais como ícones confusos, falta de padronização dos botões, navegação confusa, etc., além de revisão do texto. Além disso, ela testa o OA com a finalidade de verificar possíveis erros. Ela precisa estar em contato com a equipe técnica e, dependendo do problema, com a equipe de domínio. Além da validação interna com a equipe de domínio (conteúdos) e a técnica (programação), a inserção deste (a) profissional permitiu a validação da usabilidade de um OA.

A Figura 3.9 apresenta o OA “Balão”, desenvolvido pela equipe RIVED-UENF e apresenta os recursos do OA. Pode-se ver que o OA “Balão” é composto de imagens, animação, botões de navegabilidade e também de parte textual.

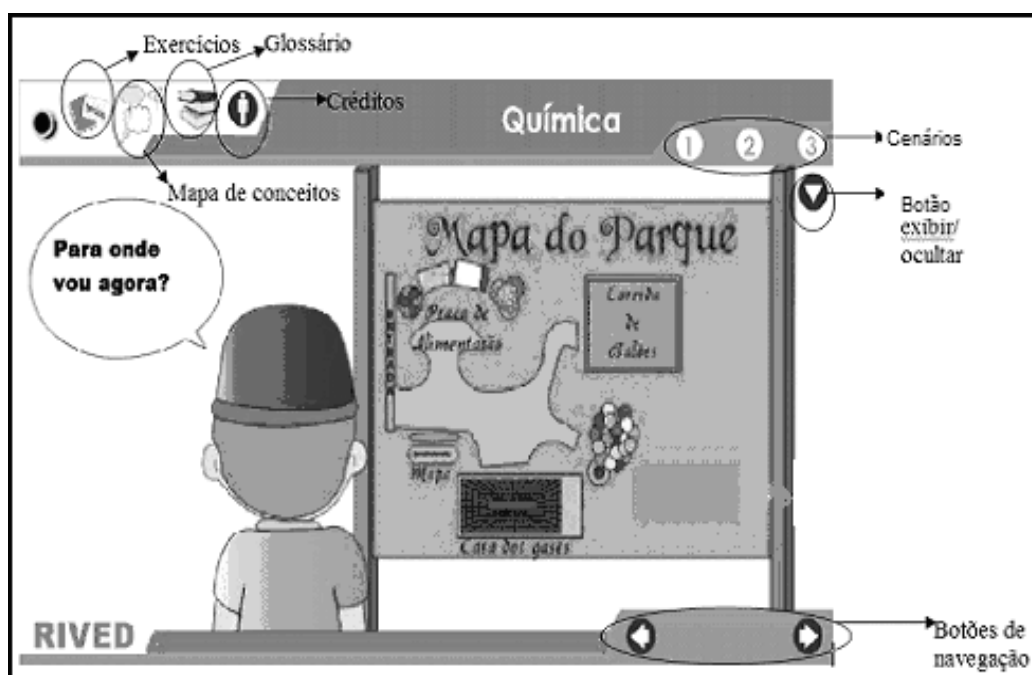


Figura 3.9. Tela do OA Balão

Fonte: Equipe RIVED-UENF

Ao analisar as fases do processo produtivo de OA da equipe RIVED-UENF pode-se verificar que elas abrangem as atividades propostas por Pressman (2006), apesar de serem mais genéricas.

3.2.1 Plataformas de desenvolvimento

Para o desenvolvimento dos OA, a equipe RIVED-UENF utiliza a ferramenta da Macromedia *Flash*, a linguagem *ActionScript* e ainda a linguagem XML. O *software Flash* possui uma interface que permite a criação de recursos gráficos, como imagens, filmes animados, importação de arquivos externos, etc. Como os OA são compostos de vários recursos multimídia, utiliza-se o *Flash*, pois ele permite fazer animações simples e complexas. O *Flash* permite fazer animações com interatividade, como gerar um filme AVI ou um executável (SANTOS, 2001). O *ActionScript* é a linguagem de script do *Macromedia Flash* MX que possibilita a manipulação dos elementos (ou objetos) que compõem um filme e a manipulação do filme como um todo. Através do *ActionScript* é possível definir o comportamento de um programa em *Flash*.

Como forma de facilitar a edição de textos dos OA, algumas equipes RIVED, inclusive a RIVED-UENF, utilizam a linguagem XML. XML é uma sigla que

representa *eXtensible Markup Language* – linguagem de marcação extensível (W3C, 2007).

A linguagem XML deriva do Padrão de Linguagem Genérica de Marcadores (SGML - *Standard Generalized Markup Language*), o qual foi estabelecido por volta de 1969 para unificar a confusão de linguagens existentes na época para a formatação de documentos para publicação.

XML foi lançada oficialmente em 1996 pelo W3C (*World Wide Web Consortium*), organização responsável pela regulamentação dos padrões da Internet. XML possui como característica marcante a combinação da capacidade e flexibilidade da SGML com a enorme aceitação da HTML (*Hypertext Markup Language*). A linguagem XML é atualmente empregada na representação de estruturas de dados estruturados e semiestruturados e seu intercâmbio na *Web* (MELLO *et al.*, 2000).

A XML parece muito com a HTML, completa com *tags*, atributos e valores. Porém, como mostra Pacheco e Kern (2001) a linguagem HTML possui um conjunto fixo de tags, enquanto a linguagem XML é autodescritiva.

3.2.2 Problemas encontrados na fase de desenvolvimento dos OA

Conforme apresentado no capítulo 1, que contém definições de OA, um OA utiliza vários recursos como imagens, animações, sons, textos, etc. Os textos permitem que um conteúdo seja transmitido ao usuário com a ajuda dos outros recursos. Nesse contexto, é fundamental que a parte textual dos OA seja de fácil manipulação tanto pela equipe que os desenvolve, como pela equipe de domínio. A Figura 3.10 apresenta o arquivo XML de um OA.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<animation>
<screens>
<!-- Textos da padronização -->
<screen index="0">
<text>
<text>Química</text>
<text>Interface 0</text>
<text>RIVED</text>
</text>
</screen>
<!-- Apresentação/Créditos -->
<screen index="1">
<text>
<text>Casa dos</text>
<text>Gases</text>
<text>Clevis Elena Rapikevicz, D.Sc.</text>
<text>coordenação tecnológica</text>
<text>Maria Cristina Canela, D.Sc.</text>
<text>coordenação pedagógica</text>
<text>Eduardo Coelho Carneiro</text>
<text>desenvolvimento</text>
<text>Gláucia Ribeiro Gonzaga</text>
<text>design gráfico e modelagem</text>
<text>Valéria Machado da Costa</text>
<text>avaliação visual e supervisão</text>
<text>ENTRAR</text>
</text>
</screen>
<!-- Textos em camadas-->
<screen index="2">
<text>
<text>Clique nos cilindros para ver as características dos gases contidos nos mesmos.</text>
<text>Neste Parque você pode conhecer a casa dos gases e os mistérios de alguns balões; pode ir à praça de alimentação e observar os diferentes comportamentos dos gases e também pode participar de uma entusiasmada comida de balões e aplicar tudo o que você já aprendeu nas aulas de química. Bom divertimento!</text>
<text>Agora você pode verificar o comportamento dos gases arrastando até três cilindros cada vez para o quadro. Depois clique em testar e observe o que acontece.</text>

```

Figura 3.10. XML do Objeto Balão

Fonte: Equipe RIVED-UENF

Esse arquivo XML externo permite que a equipe de domínio e até mesmo os usuários alterem a parte textual dos OA sem a necessidade de ter acesso ao código-fonte de um OA (um arquivo em *Flash*). Mas alguns problemas ocorrem quanto ao uso do arquivo XML. A equipe de domínio, responsável pelo conteúdo de um OA e, portanto, a mais apropriada para criar e alterar a parte textual do arquivo, possui dificuldades quanto a criação e edição do arquivo XML.

Os problemas estão relacionados à manipulação do arquivo pela equipe de domínio e por falhas do arquivo XML em si. Quanto às alterações por parte da equipe de domínio, o problema é a falta de uma ferramenta adequada que permita a criação e manipulação do arquivo XML. A criação e manipulação do arquivo XML são atividades feitas pela equipe técnica que copia as informações do Roteiro, descrito na seção 3.2, e cola no arquivo XML. De acordo com o processo produtivo da equipe RIVED-UENF, após a fase de validação, se um OA apresentar erros, retorna-se às fases anteriores para que as correções sejam feitas. Dessa forma, o trabalho de criar e modificar o arquivo XML recai novamente para a equipe técnica. Isso causa desperdício de tempo, uma vez que a equipe de domínio é quem faz as

correções de conteúdo. Portanto, a criação do arquivo XML e as modificações após a fase de validação poderiam ser feitas pela equipe de domínio. Para isso, é necessário que a equipe de domínio utilize uma ferramenta que permita manipular o arquivo XML, de forma fácil e segura, sem causar erros no conteúdo dos OA.

A forma como a equipe RIVED-UENF utiliza o arquivo XML ocasiona problemas como:

- A equipe RIVED-UENF não possui uma forma de validar o documento XML, ou seja, não possui um esquema que permita descrever os objetos que podem aparecer no arquivo XML (elementos, atributos). Um documento XML é válido se houver um esquema associado a ele;
- O arquivo XML não possui uma organização. Isso permite que dentro da mesma equipe existam arquivos estruturados de forma diferente. As estruturas são diferentes porque cada OA possui um conteúdo em particular;
- A hierarquia das informações do arquivo XML de um OA não é a mesma utilizada por outros OA. Por exemplo, o local onde se encontram os *helps* de contexto (auxiliam na navegação do OA) difere de um OA para outro. É necessário estruturar o arquivo XML de modo que ele possua partes em comum com outros OA e ao mesmo tempo permita ter sua particularidade em relação ao conteúdo a ser abordado;
- Apesar da XML ser uma linguagem de fácil compreensão, ela possui *tags* (elementos), que se apagadas ou alteradas de modo equivocado geram erro como, por exemplo, os textos não serem visualizados no OA. Uma das dificuldades encontradas é permitir que qualquer pessoa altere a parte textual dos OA sem atrapalhar a sua funcionalidade, pois, se uma *tag* é deletada ou alterada no arquivo XML erroneamente, os textos não são exibidos corretamente por um OA;
- Outro problema é a existência de *tags* com conteúdo que não está sendo utilizado no OA, ou seja, é um conteúdo que está no arquivo XML sem exercer função alguma, mas que, se retirado, gera erro porque o modo de indexação do objeto especifica a posição da *tag*. Dessa forma, se um elemento é excluído, o texto no objeto será trocado por outro texto da posição anterior. Essa forma de utilização da XML está inadequada, uma vez que o arquivo XML deveria ser utilizado de forma não-posicional;
- Os textos que necessitam de subscrito ou sobrescrito, como as fórmulas químicas, não aparecem de forma correta nos objetos, isso porque o

arquivo XML, da forma como está estruturado, serve apenas para a descrição dos dados.

O Quadro 3.1 sintetiza os problemas relacionados ao uso do arquivo XML pela equipe de domínio da equipe RIVED-UENF e a forma como o arquivo XML é estruturado.

Problemas	Descrição
O documento XML não possui validação.	A equipe RIVED-UENF não possui uma forma de validar o documento XML. Um documento XML é dito válido se houver um esquema associado a ele. Um elemento fundamental pode ser apagado ou omitido, causando erros na exibição e funcionamento de um OA. Com a validação é possível saber que o arquivo XML apresenta erros.
A hierarquia dos arquivos XML não é padronizada.	A hierarquia de um arquivo XML de um OA não é a mesma utilizada por outros OA. Isso ocasiona uma confusão quanto a ordem de exibição dos conteúdos.
As <i>tags</i> podem ser apagadas equivocadamente.	As <i>tags</i> podem ser apagadas e/ou alteradas de modo equivocado, ocasionando erro no conteúdo a ser exibido em um OA.
Os documentos XML são diferentes.	Dentro da mesma equipe (UENF) existem arquivos XML diferentes. A falta de padronização dificulta o entendimento das informações.

Quadro 3.1. Descrição dos problemas dos OA

Fonte: Adaptado de Santos, Rapkiewicz e Wives (2008a, p.7)

Após a descrição dos problemas encontrados no arquivo XML, que contém as informações dos OA da equipe RIVED-UENF, pode-se concluir que são necessárias modificações, de modo que ele se torne padronizado, organizado e flexível. Quanto às dificuldades da equipe de domínio em editar os metadados da parte textual dos OA, é fundamental analisar se os editores XML existentes atendem as necessidades ou se será necessário desenvolver uma ferramenta que resolva estes problemas.

Além da XML existem outras linguagens, derivadas da mesma. Cabe analisar se essas linguagens serviriam para solucionar os problemas em questão.

4. REPRESENTAÇÃO DA INFORMAÇÃO

Por mais parecidos que sejam os conceitos de dado e informação, cada um deles tem um significado bem particular mesmo estando relacionados. De acordo com Setzer (1999), dado é uma sequência de símbolos quantificados ou quantificáveis. Segundo Silva, Ferreira e Borges (2002) a informação é o dado com significado, faz sentido e é compreendido por alguém. O acúmulo de dados não significa informação, e o acúmulo de informação não representa conhecimento, pois cada qual tem a sua distinção e seu papel no ciclo (SILVA, FERREIRA e BORGES, 2002). Pressman (2006) define dado como “informação bruta, coleções de fatos que precisam ser processados para serem significativos, enquanto a informação é derivada da **associação de fatos em um dado contexto**” (grifo nosso).

Os OA possuem diversas informações, que vão desde o conteúdo da área de domínio até os recursos que o compõem, como imagens, botões, sons e etc. Pode-se dizer que esses recursos constituem a estrutura de informação dos OA. A estrutura de informação de um OA contém elementos principais que o descrevem como um todo, e outros elementos que descrevem os recursos existentes em cada uma de suas cenas (SANTOS, RAPKIEWICZ e WIVES, 2008a).

No contexto dessa dissertação, a estrutura de informação é útil para estruturar, organizar e gerenciar todas as informações pertinentes aos OA. Parafraseando Pressman, os **fatos** que precisam ser associados, no caso dos OA, são seus vários elementos constituintes (som, imagens, textos, etc.). O **contexto** no qual esses elementos são associados são as cenas.

Para encontrar uma linguagem que represente a estrutura de informação fez-se necessário conhecer as características de algumas linguagens de marcação. A análise das linguagens possibilita que seja utilizada uma linguagem adequada para representação da estrutura de informação.

4.1 Linguagens de marcação

De acordo com Bax (2001), as linguagens de marcação permitem a construção de padrões públicos e abertos que estão sendo criados na tentativa de maiores avanços no tratamento da informação; elas minimizam o problema de transferência de um formato de representação para outro e liberam a informação das tecnologias de informação proprietárias. Existem dois tipos de marcação: a

marcação procedimental e a marcação descritiva. Na marcação procedimental cada código do sistema indica o procedimento a ser seguido para a apresentação do texto ao usuário (MARCHAL 2000 *apud* BAX 2001). Já na marcação descritiva utilizam-se marcas (ou *tags*) para qualificar cada objeto do texto, cumprindo um primeiro passo para transformá-los em informação tratável por computador (BAX, 2001). Serão apresentadas a seguir algumas linguagens e suas características.

4.1.1 SGML

SGML (*Standard Generalized Markup Language*) é um padrão internacional para a definição de dispositivos e sistemas independentes de métodos de representação de textos em formato eletrônico. SGML é um padrão internacional para a descrição de texto de marcação eletrônico. Mais exatamente, o SGML é uma metalinguagem, isto é, um meio de descrever uma linguagem formal, neste caso, uma linguagem de marcação (SGML, 2005).

Para Souza e Alvarenga (2004) o padrão SGML, é na verdade, uma metalinguagem, ou seja, uma linguagem para descrever outras linguagens. O padrão SGML é baseado na idéia de que documentos contêm estrutura e outros elementos semânticos que podem ser descritos sem que se faça referência à forma com que estes elementos serão exibidos.

A SGML pode ser chamada de “língua mãe” e é usada para descrever tipos diferentes de documentos em muitas áreas da atividade humana (ALMEIDA, 2002).

4.1.2 HTML

A linguagem HTML (*Hypertext Markup Language*) é usada basicamente para a apresentação de conteúdo. HTML é uma linguagem para publicação de hipertexto na *World Wide Web*. É um formato não proprietário baseado em SGML, e pode ser criado e tratado por uma grande quantidade de ferramentas, inclusive editores de texto simples (W3C, 2007). Inicialmente foi concebida como uma solução para a publicação de documentos científicos em meios eletrônicos, que ganhou popularidade e se tornou padrão para a Internet.

Consiste de uma aplicação específica do SGML utilizada na Internet, que define um tipo de documento simples, com marcações fixas projetadas para uma classe de relatórios técnicos de uso comum em escritórios como, por exemplo,

cabeçalhos, parágrafos, listas, ilustrações e algumas possibilidades para hipertexto e multimídia (ALMEIDA, 2002). É um padrão muito utilizado na Internet.

4.1.3 XML

XML foi lançada oficialmente em 1996 pelo W3C (*World Wide Web Consortium*), organização responsável pela regulamentação dos padrões da Internet. XML possui como característica marcante a combinação da capacidade e flexibilidade da SGML com a enorme aceitação da HTML. A linguagem XML é atualmente empregada na representação de estruturas de dados estruturados e semiestruturados e seu intercâmbio na *Web* (MELLO *et al.*, 2000).

XML fornece subsídios para a definição de marcadores que permitem agregar ao conteúdo do documento algum significado ou informação referente ao seu conteúdo. Em especial, é possível utilizar a XML para inserir metadados (dados que descrevem dados) em um documento, o que possibilita não apenas a apresentação dos dados para seres humanos, como também o processamento de tais informações por aplicações (SANTANCHÈ e TEIXEIRA, 1999).

A XML parece muito com a HTML, completa com *tags*, atributos e valores. Porém, HTML exhibe dados e é voltada para apresentação dos mesmos. A XML descreve dados e dá ênfase ao significado deles.

Para validar um documento XML pode ser utilizado o XML *Schema*, que é uma recomendação oficial do W3C. O propósito de um XML *Schema* é definir os blocos de construção permitidos em um documento XML. Também é chamada de XML *Schema Definition* (XSD) (W3SCHOOLS, 2007). O XML *Schema* é a evolução do DTD (*Document Type Definition*), que é essencialmente uma gramática que especifica as *tags* e a estrutura de um documento XML. A DTD é uma definição da XML que permite a descrição e a caracterização da estrutura e do conteúdo de um determinado documento, descrevendo exatamente as marcações válidas, a ordem em que devem aparecer, as marcas que podem aparecer dentro de outras, os atributos necessários e os valores que podem ter (TOLENTINO e MOITA, 2002).

Como o XML *Schema* é uma evolução do DTD optou-se por utilizar o XML *Schema*.

4.1.4 YAML

A linguagem YAML possui dois nomes - aparentemente contraditórios – que têm sido atribuídos para o acrônimo YAML, sendo eles : “*Yet Another Markup Language*” e o mais recursivo “*YAML Ain’t a Markup Language*”. A lógica disso é que o YAML oferece grande parte do que as linguagens de marcação oferecem, mas com uma notação muito leve (FONSECA e SIMOES, 2007).

O YAML é um formato de serialização⁵ de dados legível por humanos, que se inspirou em conceitos de linguagens como XML, C, *Python*, *Perl* e também no formato do correio eletrônico especificado no RFC 2822 (FONSECA e SIMOES, 2007).

YAML é o resultado de estudos a partir de XML e outras tecnologias. A Figura 4.1 mostra como YAML possui maneiras alternativas de apresentar o mesmo tipo de dados.

```

--- # Filmes favoritos, formato bloco
- Casablanca
- Spellbound
- Notorious
--- # Lista de compras, formato inline
[milk, bread, eggs]

--- # Bloco
name: John Smith
age: 33
--- # Inline
{name: John Smith, age: 33}

```

Figura 4.1. Representação de dados em blocos e linhas

Fonte: Fonseca e Simões (2007)

Os objetivos do modelo YAML são:

- ser facilmente legível pelo homem;
- permitir dados portáveis entre linguagens de programação;
- ser um modelo consistente para apoiar ferramentas genéricas;
- ser expressivo e extensível;
- ser de fácil implementação e utilização.

⁵ Serialização é um mecanismo da linguagem Java que permite o armazenamento de objetos e todos os demais objetos relacionados (recursivamente) em um arquivo do sistema operacional (WERNER *et al.*, 2000).

4.1.5 JSON

O JSON significa *JavaScript Object Notation*. JSON é um subconjunto da notação de objeto de *Javascript* e possui um formato de texto para a serialização de dados estruturados. JSON foi desenhado com o objetivo de ser simples, portátil e textual (FONSECA e SIMOES, 2007). É tipicamente usado em ambientes onde o tamanho do fluxo de dados entre o cliente e o servidor é de extrema importância (daí seu uso por *Google*, *Yahoo*, etc., os quais servem milhões de usuários), onde a fonte dos dados pode ser explicitamente confiável, e onde a perda dos recursos de processamento XSLT⁶ no lado cliente para manipulação de dados ou geração da interface não é uma consideração. JSON é um formato de texto completamente independente de linguagem, pois usa convenções que são familiares às linguagens C, incluindo C++, C#, Java, *JavaScript*, *Perl*, *Python* e muitas outras (JSON, 2008).

```
{
  "type": "menu",
  "value": "File",
  "items": [
    {"value": "New", "action": "CreateNewDoc"},
    {"value": "Open", "action": "OpenDoc"},
    {"value": "Close", "action": "CloseDoc"}
  ]
}
```

Figura 4.2. Exemplo JSON

Fonte: Fonseca e Simões, 2007

O exemplo da Figura 4.2 poderia ser escrito em apenas uma linha. Apesar de a sintaxe do JSON ser parecida com a YAML, nela a indentação e o uso de espaços em branco é totalmente opcional (FONSECA e SIMOES, 2007).

4.1.6 OGD

OGDL (*Ordered Graph Data Language*) é um dos inúmeros esforços no sentido de desembaraçar a sintaxe XML. É uma linguagem de descrição de dados em grafos ordenados. OGD é um formato textual que representa árvores ou grafos de dados, onde os nós são *strings* e as bordas são espaços ou indentação (OGDL, 2005). Ela representa nós de informação e suas relações. O grafo consiste de nós e

⁶ XSLT- é uma linguagem para transformar documentos XML em outros documentos XML. XSLT transforma uma árvore de origem XML em uma árvore de resultado XML (W3SCHOOLS, 2008).

ligações entre os nós. Ambos podem ser rotulados para indicar a semântica das relações.

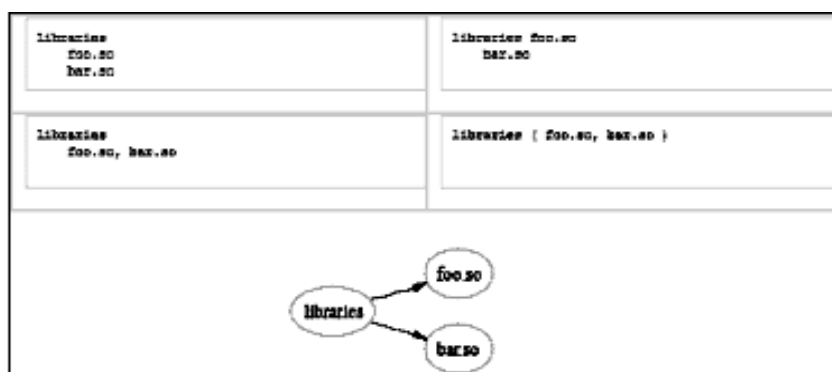


Figura 4.3. Exemplos OGDL

Fonte: <http://ogdl.sourceforge.net/examples.htm>

A Figura 4.3 apresenta quatro exemplos equivalentes e que produzem o mesmo grafo. Os principais objetivos do OGDL são simplicidade, legibilidade e facilidade de acesso.

A criação da estrutura padrão requereu uma análise das linguagens de marcação, a fim de verificar se XML é realmente adequada. O Quadro 4.1 apresenta o resultado das linguagens descritas anteriormente.

Linguagens	A linguagem é adequada para representar a estrutura de informação de um OA? Porquê?
SGML	Não. A SGML é uma metalinguagem, isto é, um meio de descrever uma linguagem formal. A linguagem é complexa.
HTML	Não. É usada basicamente para a apresentação de conteúdo. É uma linguagem para formatação da informação.
XML	Sim. Como a SGML, XML também permite a criação de modelos de documentos. Por ser subconjunto do SGML, seu uso é mais fácil. É uma linguagem de marcação extensível, utilizada na representação de dados. Descreve os conteúdos de documentos. Os documentos XML são autodescritivos.

YAML	Não. A mistura de elementos com o texto é algo que dificilmente se consegue com YAML. O YAML não se aproxima ao poder que a XML tem para anotar documentos. É utilizada para serialização de dados.
JSON	Não. Sendo um subconjunto funcional do YAML, o JSON apresenta praticamente o mesmo poder expressivo e capacidade de anotação (quase nula) do YAML. A definição de dados em <i>JavaScript</i> não é o objetivo desse trabalho.
OGDL	Não. É uma linguagem de descrição de dados em grafos ordenados. Seria útil apenas como um modelo. Não há um validador de documentos.

Quadro 4.1. Análise de linguagens

Ao analisar as características de cada linguagem, pode-se notar que a XML é a mais adequada, por ser uma linguagem de marcação extensível, autodescritiva, permitindo definir e compor elementos específicos de um domínio.

4.2 Padrões de metadados para Objetos de Aprendizagem

Atualmente existem algumas outras organizações com interesses e argumentos comuns que desenvolvem especificações para *e-learning*, como por exemplo, IMS (*IMS Global Learning Consortium*), AICC (*Aviation Industry CBT Committee*), ADLNet (*Advanced Distributed Learning*) e W3C (*Extensible Markup Language*). Há vários padrões de metadados para OA, como por exemplo o LOM (*Learning Object Metadata*), o SCORM (*Sharable Content Object Reference Model*) e o DCMI (*Dublin Core Metadata Initiative*), os quais são apresentados a seguir.

4.2.1 LOM

Organizações como IEEE (*Institute of Electrical and Eletronic Engineer's*) e o IMS se esforçaram para o desenvolvimento de padrões para a descrição de objetos de aprendizagem e propuseram o padrão *Learning Object Metadata*, ou seja, Metadados para Objetos de Aprendizagem (LOM) como um padrão. O LOM é um padrão que segue os propósitos genéricos de metadados, possibilitando que objetos educacionais desenvolvidos, organizados e armazenados neste padrão possam ser

recuperados. Ainda pode referenciar outros objetos através de blocos, os quais podem ser combinados sequencialmente permitindo a construção de unidades educacionais maiores ou objetos compostos (SCHEER e GAMA, 2004).

De acordo com Mallard (2004), o padrão LOM destina-se a facilitar a utilização de objetos de aprendizagem por processos automatizados e pessoas e sua especificação define nove categorias de metadados, que servem para a descrição dos OA, sendo elas:

- *General* (Geral): Identificador, título, idioma, descrição e palavras-chave;
- *Life Cycle* (Ciclo de vida): *status*, versão, tipo de contribuição, entidades que contribuíram, data da contribuição;
- *Meta MetaData* (Metadados): metadados para descrever os metadados usados para o OA;
- *Technical* (Técnica): formato, tamanho, localização, tipo de tecnologia, nome da tecnologia, duração;
- *Educational* (Educativo): informações sobre os objetivos educacionais de um OA, como interatividade, dificuldade, tipo de usuário final, etc.;
- *Rights* (Direitos): uso comercial e propriedade de um OA;
- *Relation* (Relações): relacionamentos com outros OA;
- *Annotation* (Comentários): informações adicionais sobre um OA;
- *Classification* (Classificação): define diferentes propósitos de um OA.

A estrutura do padrão LOM é definida em um formato hierárquico, o qual apresenta vários níveis de detalhamento. Na Figura 4.4 pode-se ver um esquema de metadados do padrão LOM.

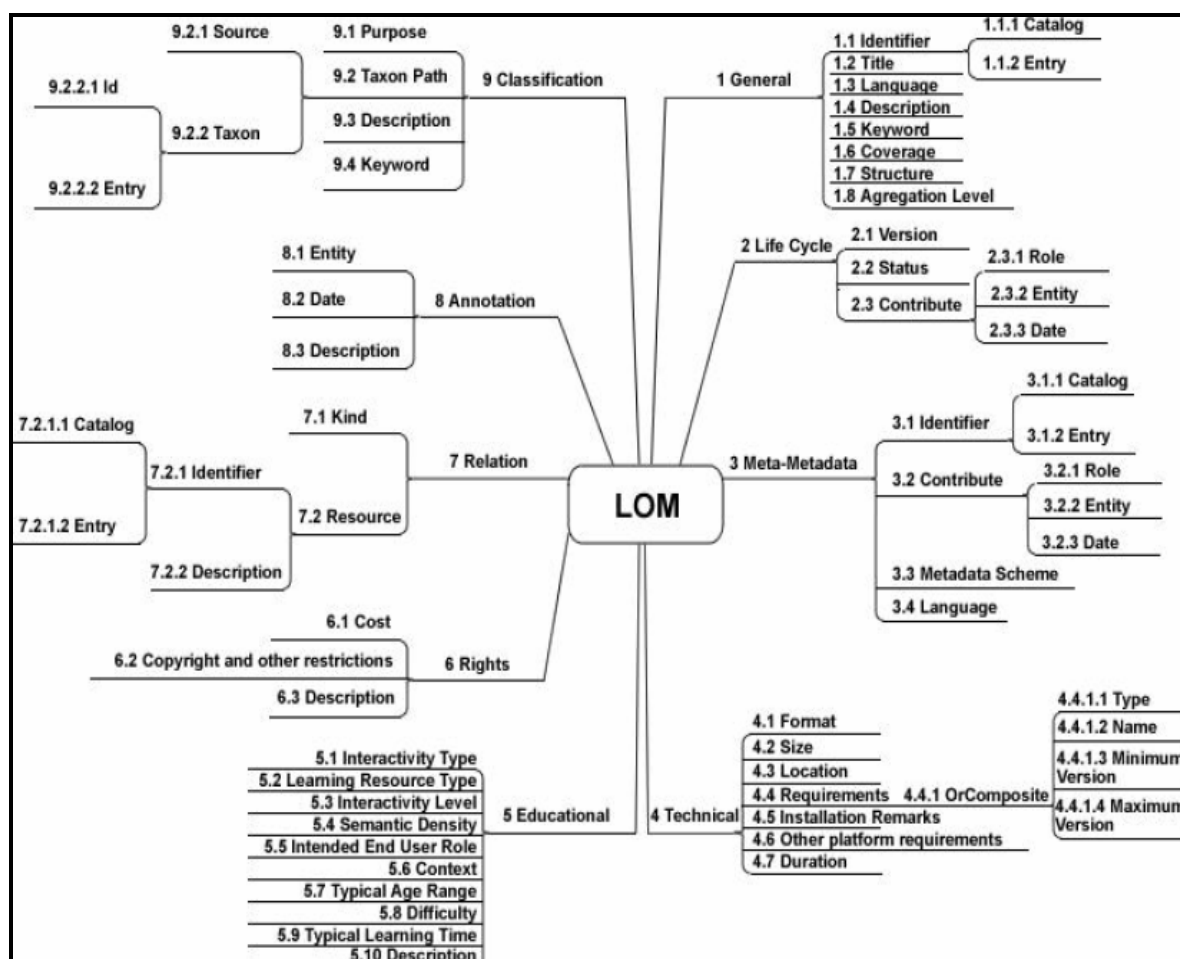


Figura 4.4. Esquema de metadados do padrão LOM

Fonte: Warpechowski (2005)

O LOM é de suma importância para organizar a estrutura de informação dos OA, pois suas categorias conseguem descrevê-los de forma bem detalhada.

4.2.2 SCORM

SCORM é um conjunto de normas e especificações adaptadas a partir de várias fontes para fornecer uma abrangente plataforma de *e-learning* que permita interoperabilidade, acessibilidade e reutilização de conteúdos de aprendizagem baseados na *Web* (ADL, 2008). Consiste em um modelo que referencia um conjunto de padrões técnicos, especificações e diretrizes desenvolvidas para atender a requisitos de alto nível para conteúdos e sistemas para aprendizagem (BRITO, PEREIRA e BRAGA, 2006).

De acordo com Dutra, Tarouco e Passerino (2008), o SCORM é formado por um conjunto de especificações publicadas na forma de livros. Nesses livros estão

compiladas recomendações acerca da organização dos conteúdos, como esses conteúdos são estruturados e como será a navegação neles.

Um material no formato SCORM contém uma estrutura e regras de apresentação desse conteúdo. Ele pode conter qualquer arquivo que possa ser exibido em um navegador, bem como atividades interativas, exercícios e questionários (VAHLICK e RAABE, 2008).

4.2.3 DCMI

O padrão DCMI (*Dublin Core Metadata Initiative*), ou *Dublin Core* (DC) pode ser definido como sendo o conjunto de elementos de metadados planejado para facilitar a descrição de recursos eletrônicos. As principais características do padrão DC são a simplicidade na descrição dos recursos, entendimento semântico universal (dos elementos), escopo internacional e extensibilidade (SOUZA, VENDRUSCULO e MELO, 2000). Ele possui um conjunto de metadados (Figura 4.5) e permite a inclusão de elementos adicionais.

Elementos	Descrição
Título	Nome dado ao recurso
Criador	Entidade originalmente responsável pela criação do conteúdo do recurso
Assunto	Tema do conteúdo do recurso. Pode ser expresso em palavras-chaves e/ou Categoria. Recomenda-se o uso de vocabulários controlados
Descrição	Relato do conteúdo do recurso. Exemplos: texto livre, sumário e resumo
Publicador	Entidade responsável por tornar o recurso disponível
Colaborador	Entidade responsável pela contribuição intelectual ao conteúdo do recurso
Data	Data associada a um evento ou ciclo de vida do recurso
Tipo	Natureza ou gênero do conteúdo do recurso. Exemplos: texto, imagem, som, dados, software
Formato	Manifestação física ou digital do recurso. Exemplos: html, pdf, ppt, gif, xls
Identificador	Referência não-ambígua (localizador) para o recurso dentro de dado contexto
Fonte	Referência a um recurso do qual o presente é derivado
Idioma	Língua do conteúdo intelectual do recurso
Relação	Referência para um recurso relacionado
Cobertura	Extensão ou escopo do conteúdo do recurso; pode ser temporal e espacial

Figura 4.5. Metadados Dublin Core

Fonte: Alves e Souza (2007)

O *Dublin Core* utiliza a linguagem XML e pode ser inserido em uma página HTML. Além disso, adota a sintaxe do RDF (*Resource Description Framework*), possui um conjunto de 15 elementos básicos que podem ser implementados livremente de acordo com as necessidades de cada usuário, sendo um padrão adotado para efetuar a interoperabilidade entre outros formatos (ALVES e SOUZA, 2007).

5. PADRONIZAÇÃO DA ESTRUTURA DE INFORMAÇÃO

Esse capítulo está organizado em três seções. A primeira seção apresenta a análise de OA de diferentes equipes RIVED na área de Química buscando identificar os elementos da estrutura de informação dos OA. A segunda seção apresenta a padronização da estrutura de informação proposta e a terceira seção apresenta a representação desta estrutura em XML.

5.1 Análise dos elementos da estrutura de informação em diferentes equipes

As equipes RIVED pesquisadas são da área de Química, sendo das seguintes universidades:

- UNESP - Universidade Estadual de São Paulo;
- UFMA - Universidade Federal do Maranhão;
- UFU - Universidade Federal de Uberlândia;
- UNIFRA - Centro Universitário Franciscano (Santa Maria, RS);
- UFC - Universidade Federal do Ceará;
- UENF – Universidade Estadual do Norte Fluminense.

Foi feito um levantamento da plataforma de *software* utilizada para desenvolvimento dos objetos, ferramentas utilizadas bem como o uso de padrão de metadados. O Quadro 5.1 sintetiza o que foi constatado.

Perguntas	UNESP	UFMA	UFU	UNIFRA	UFC	UENF
Sistema Operacional utilizado	Windows	Windows	Windows	Windows	Windows	Windows
Usa padrão de metadados no próprio objeto	Não	Não	Não	Não	Não	Não
Ferramenta usada para edição de metadados	Não possui ferramenta específica. Usa bloco de notas.	Não possui ferramenta específica. Usa bloco de notas e <i>DreamWeaver</i> .	Não possui ferramenta específica. Usa <i>Microsoft Word</i> .	Não possui ferramenta específica. Usa bloco de notas.	Não possui ferramenta específica. Usa bloco de notas.	Não possui ferramenta específica. Usa bloco de notas ⁷

⁷ Situação no momento da análise do processo de trabalho das equipes. No ano de 2009 passou a utilizar o editor desenvolvido nesta dissertação.

Ferramenta de autoria e linguagens utilizadas	HTML, <i>Action Script</i> IDE: <i>Adobe Macromedia Flash</i> , XML	HTML, <i>Action Script</i> IDE: <i>Adobe Macromedia Flash</i> , XML Softwares de apoio: <i>3d Studio Max</i> , <i>PhotoShop</i> , <i>Fireworks</i>	<i>ActionScript</i> IDE: <i>Adobe Macromedia Flash</i> , XML, <i>ActionScript</i>	<i>ActionScript</i> IDE: <i>Adobe Macromedia Flash</i>	HTML <i>ActionScript</i> IDE: <i>Adobe Macromedia Flash</i>	<i>ActionScript</i> , IDE: <i>Adobe Macromedia Flash</i> , XML
---	--	--	--	---	---	---

Quadro 5.1. Informações obtidas das equipes RIVED

Conforme apresenta o Quadro 5.1, o sistema operacional utilizado é o *Windows*. As linguagens utilizadas no desenvolvimento dos OA são XML, HTML e *ActionScript (Macromedia Flash)*. Nenhuma delas utiliza Java.

Conforme já destacado, um OA é composto de várias mídias (texto, imagens, som, vídeo). Considerando-se o foco desta dissertação, foi analisado como cada equipe trata a parte textual dos OA. Os textos são informações como explicação de conteúdo, fala de personagens, palavras existentes nos cenários⁸ e informações para ajudar o usuário a entender o OA. O Quadro 5.2 apresenta as quatro formas encontradas.

Equipe	Texto no código-fonte compilado	Texto no código-fonte interpretado	Texto em uma estrutura separada	Texto armazenado em forma mista
UENF			X	
UNESP	X			
UFMA				X
UFU			X	
UNIFRA	X			
UFC	X			

Quadro 5.2. Comparativo das formas de representação de textos utilizadas pelas equipes RIVED Química

Fonte: Santos *et al.* (2008)

⁸ Cenários representam as várias telas e os diferentes elementos integrantes de cada situação em particular.

No primeiro modelo que aparece no Quadro 5.2, o texto é inserido junto com o código-fonte compilado, formando um arquivo único e sua visualização depende do sistema que irá executá-lo. No segundo modelo o texto é inserido no código-fonte interpretado, misturado aos códigos de linguagem, onde cada objeto possui estruturas diferentes para cada código-fonte. No terceiro modelo os textos são inseridos em arquivos externos ao objeto, organizados de forma que o objeto localize o trecho exato que será mostrado na tela a cada momento. E no quarto modelo os textos são armazenados de forma mista, com partes armazenadas juntamente com o código-fonte e outras em um arquivo externo. De acordo com Sebesta (2000), um código-fonte compilado trata-se de uma tradução de um programa escrito em linguagem de alto nível, para um programa escrito em linguagem de máquina equivalente, enquanto no interpretado o programa fonte é aos poucos convertido em resultados por um programa denominado interpretador. O interpretador atua como uma máquina virtual para a linguagem, buscando e executando as instruções do programa.

Nos casos em que a parte textual está separada do código-fonte, as equipes utilizam arquivos XML. Tais textos são carregados para um vetor usando *ActionScript*. No momento adequado (necessidade de apresentação na tela, conforme a navegação feita pelo usuário), tais elementos são atribuídos a elementos do *Flash* para que possam ser exibidos na tela. Na prática, portanto, as características e o potencial da XML não são utilizados, de forma que qualquer arquivo texto externo ao código-fonte poderia ser usado, desde que houvesse correspondência do conteúdo do arquivo com a estrutura da matriz utilizada para armazenar os textos. Entre as características da XML desperdiçadas, destaca-se a ausência de um XML *Schema* ou DTD correspondente que permitiria validar o arquivo para evitar erros de apresentação nos objetos. Como não se usa XML *Schema* ou DTD, não há regras para inserção e remoção de elementos da estrutura de informação.

Outro problema identificado foi a edição do arquivo XML. Nenhuma equipe possui uma ferramenta específica para edição, recorrendo a *softwares* como o *Dreamweaver* e *Notepad*. Não foi constatado o uso de padrões na estruturação dos objetos. Sequer os editores que tratam a árvore XML como *Oxygen*, *Altova* e *XML Notepad* são usados. A presença de padrões é limitada à existência de um formulário com preenchimento manual no qual constam alguns elementos, tais como

nome da instituição, autores, título do OA, área de conhecimento, nível de ensino, objetivos, etc., do padrão LOM. Um exemplo completo de formulário de cadastramento encontra-se no Anexo 1. Esse formulário é usado para envio dos objetos para o MEC para fins de catalogação em um repositório.

A Figura 5.1 mostra o OA “Balão” da equipe RIVED-UENF. No Anexo 2 são apresentados os OA de outras equipes RIVED de Química e trechos de seus respectivos arquivos XML.

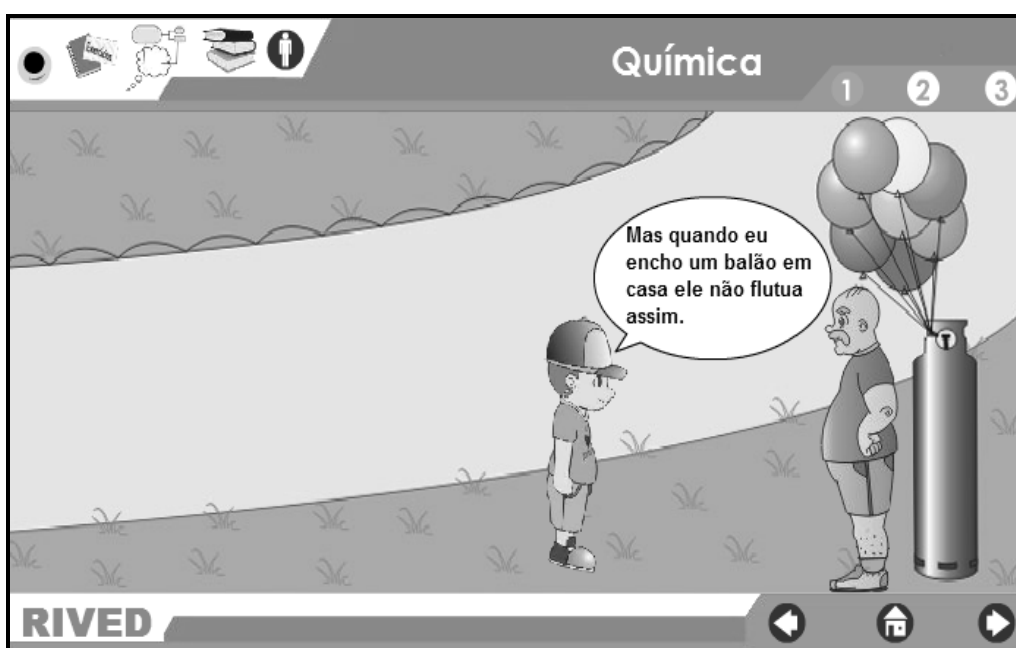


Figura 5.1. OA “Balão”

Fonte: equipe RIVED-UENF

A Figura 5.2 apresenta um trecho da parte textual do OA Balão. Observa-se que a frase sublinhada em negrito da Figura 5.2 corresponde à frase que aparece na caixa de diálogo da Figura 5.1.

```

<!-- Balão -->
<screen index="8">
  <text>
    <!-- Cenário 0 -->
    <text>Para onde vou agora?</text>
    <!-- Cenário 1 -->
    <!-- Tela 1 -->
    <text>Nossa, esse balão voa...</text>
    <text>Não é bem voar, ele flutua por causa do gás em seu interior.</text>
    <text>Mas quando eu ancho um balão em casa ele não flutua assim. </text>
    <text>É porque o gás que você sopra tem propriedades diferentes deste gás que estou utilizando.</text>
    <text>Vamos até a casa dos gases para que você entenda melhor o que estou lhe falando.</text>
    <!-- Tela 2 -->
    <text>Aqui nós temos 12 gases diferentes.</text>
    <text>Cada um possui características específicas.</text>
    <text>Pode ir olhar os cilindros se desejar.</text>
    <!-- Tela 3 -->
    <!-- Tela 4 -->

```

Figura 5.2. Parte do arquivo XML do OA “Balão”

Fonte: equipe RIVED-UENF

A estrutura de informação dos OA da equipe RIVED-UENF é formada por blocos de informações referentes a cada cenário, distinguidos através de índices (Figura 5.2) e ainda possui blocos de informações que pertencem (ou podem pertencer) a todos os cenários como botões, *helps*, etc. Nota-se que outras equipes montam a estrutura de forma diferente. Neste caso, os blocos com informações referentes a cada cenário contêm também as informações que são gerais, como os botões e os *helps*. Ou seja, isso gera uma duplicidade de informações, o que seria o lado negativo. No entanto, o aspecto positivo é que se um bloco referente a um cenário for excluído isso não danificará as informações pertencentes a outros blocos. A forma como o arquivo XML da equipe RIVED-UENF está estruturado não permite essa flexibilidade, ou seja, se o bloco que possui essas informações gerais, como os botões e *helps*, for excluído, então todos os blocos perderão informação. Analisando este problema, a estrutura de informação proposta nesse projeto (seção 5.2) seguiu o exemplo utilizado pelas outras equipes. Mesmo havendo a questão da duplicidade, isso permite que a estrutura se torne mais flexível.

Ao analisar os OA percebe-se algumas diferenças. Por exemplo, alguns possuem botões de explicação e outros não. O mesmo acontece com as opções de navegação e com *help* (geral e de contexto). O Quadro 5.3 apresenta algumas diferenças de recursos entre os OA analisados e demonstra que, apesar de alguns recursos serem comuns, a forma como são apresentados difere. A seção ‘Créditos’, por exemplo, é apresentada de formas diferentes.

Recursos	Objetos de Aprendizagem					
	Alimentos - UFMA	Abundância - UNESP	Espontaneidade - UFU	Misturas - UNIFRA	Dentro da Lei - UFC	Balão - UENF
Botões de navegação nomeados	Não	Sim	Não	Sim	Não	Sim - alguns
Créditos	Sim - somente no fim do OA	Sim - somente no início do OA	Sim – no início, mas depois desaparece	Não	Sim - permanente	Sim - permanente
Navegação linear	Sim	Não	Sim	Sim	Sim	Não
Helps de contexto	Não	Não	Sim	Não	Não	Sim
Botão de explicação de funcionamento	Sim - sempre presente	Não	Sim - conforme o uso do OA.	Sim - nos exercícios	Sim - nas atividades	Sim
Textos sobrepostos (planos diferentes)	Não	Não	Sim – nas navegações e conteúdos	Não	Sim - alguns	Sim - alguns

Quadro 5.3. Comparativo dos recursos dos OA

Fonte: Santos *et al.* (2008)

Após a análise dos recursos dos OA e das formas como os textos são representados, pôde-se concluir que falta uma padronização. A padronização é fundamental para permitir o relacionamento entre os elementos e conteúdos dos mesmos e a reusabilidade. A padronização facilita a edição e uso da informação, uma vez que as informações utilizadas na forma de texto (explicações, falas de personagem, help, etc.) são comuns a todos os OA ainda que, a forma de apresentá-los possa ser distinta. O mesmo acontece no caso dos metadados utilizados para catalogação, os quais seguem inclusive o padrão LOM.

A solução proposta para esses problemas é a padronização da estrutura de informação dos objetos incorporando os metadados utilizados na catalogação e toda as informações textuais. No âmbito dessa dissertação a fase de distribuição dos objetos através da disponibilização em repositórios não foi contemplada, por fugir ao escopo do que é tratado (processo de desenvolvimento). No entanto, considerou-se pertinente já incluir esta parte na padronização da estrutura de informação de forma que o modelo que representa a mesma esteja completo.

5.2 Especificação da estrutura de informação

Esta seção descreve a estrutura de informação padrão usada para estruturar as partes textuais dos OA, bem como os metadados do padrão LOM utilizados no cadastramento em repositório. Optou-se pelo uso do LOM porque os metadados dele, divididos em nove categorias, completam a estrutura de informação padrão. O SCORM permite empacotar conteúdo educacional e metadados e a integração entre repositórios e Sistemas de Gerenciamento de Aprendizagem (*Learning Management Systems-LMS*), mas esse não é o objetivo dessa dissertação. Além disso, os metadados utilizados no SCORM podem ser descritos pelo padrão LOM. Outra possibilidade teria sido o uso do *Dublin Core*, o qual possui um conjunto de elementos capazes de abstrair informações importantes dos objetos. Foi considerado no entanto que o LOM possui metadados que são mais adequados para a descrição de OA.

A estrutura de informação proposta foi representada através de um diagrama de classes usando UML (Figura 5.3). A sustentação para essa forma de representação encontra-se em Braganholo e Heuser (2001), os quais enfatizam que o XML *Schema* descreve um documento XML instância e um Diagrama de Classes UML pode originar vários XML *Schema* correspondentes.

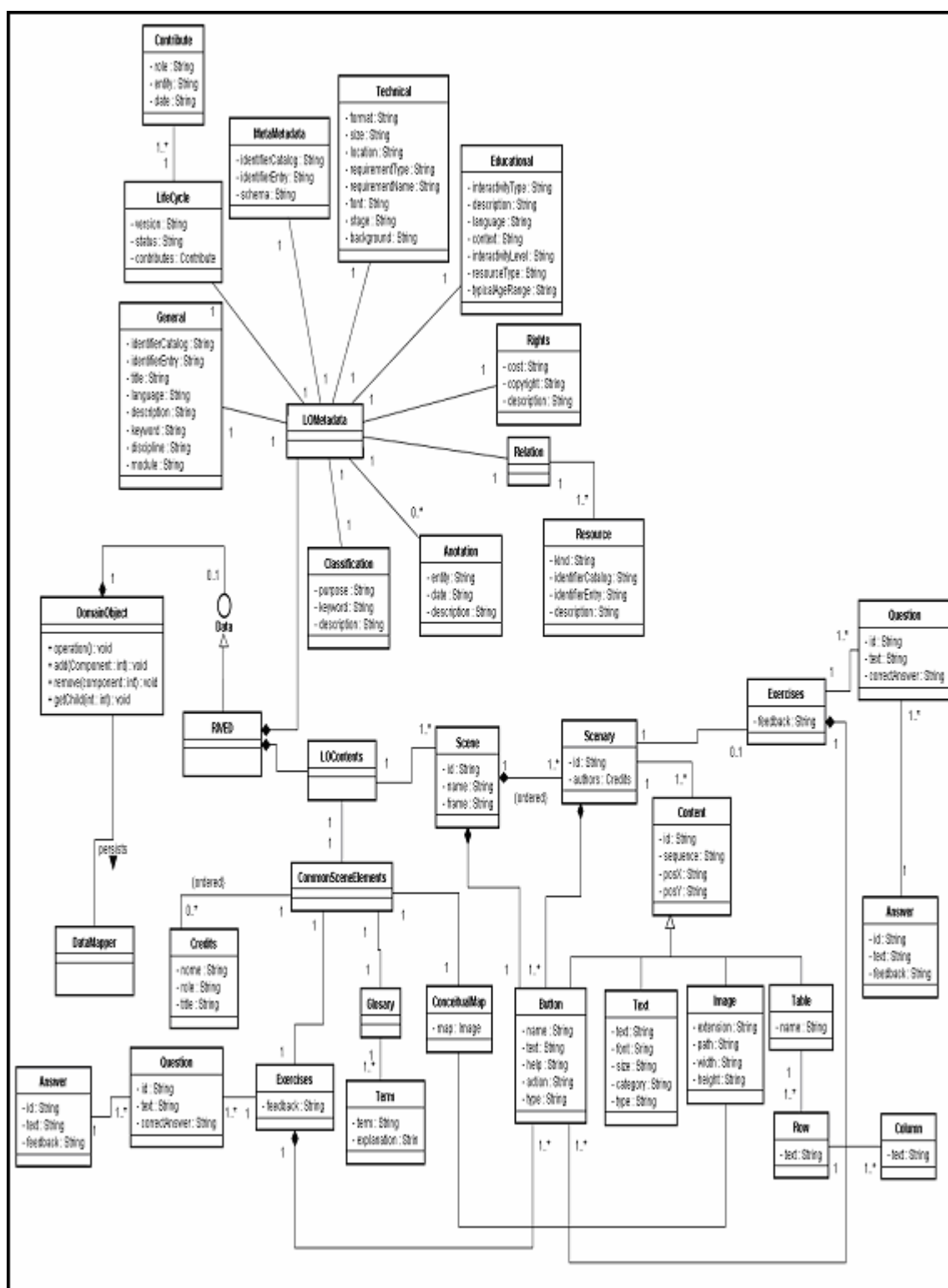


Figura 5.3. Diagrama de Classes da estrutura de informação proposta

Fonte: adaptado de Santos et al.(2008)

A estrutura de informação representada pelo diagrama de classes contém três elementos principais, dois que descrevem um OA como um todo (*LOMetadata* e *CommonSceneElements*) e uma parte que trata as cenas de um OA (*Scene*). As informações contidas nos elementos *LOMetadata* e *CommonSceneElements* são

gerais, enquanto as que pertencem ao elemento *Scene* são específicas de uma cena, uma vez que um OA pode ter uma ou várias cenas, como mostra o Diagrama de Classes.

Visando melhor descrever o diagrama, a seguir o mesmo é apresentado em duas partes. A Figura 5.4 apresenta os metadados do LOM que estão representados pelas classes:

- Geral (*General*): informações que descrevem um OA como um todo;
- Direitos (*Rights*): trata das condições de uso dos OA e direitos da propriedade intelectual;
- Educacional (*Educational*): características educacionais e pedagógicas de um OA;
- Técnica (*Technical*): características técnicas e exigências de um OA;
- Meta Metadado (*MetaMetadata*): informações sobre o próprio metadado;
- Ciclo de Vida (*LifeCycle*): características relacionadas a história e o estado atual do OA;
- Relação (*Relation*): define a relação entre um OA e outros OA correspondentes;
- Anotação (*Annotation*): fornece comentários e informações adicionais ao OA;
- Classificação (*Classification*): descreve um OA em relação a um sistema de classificação.

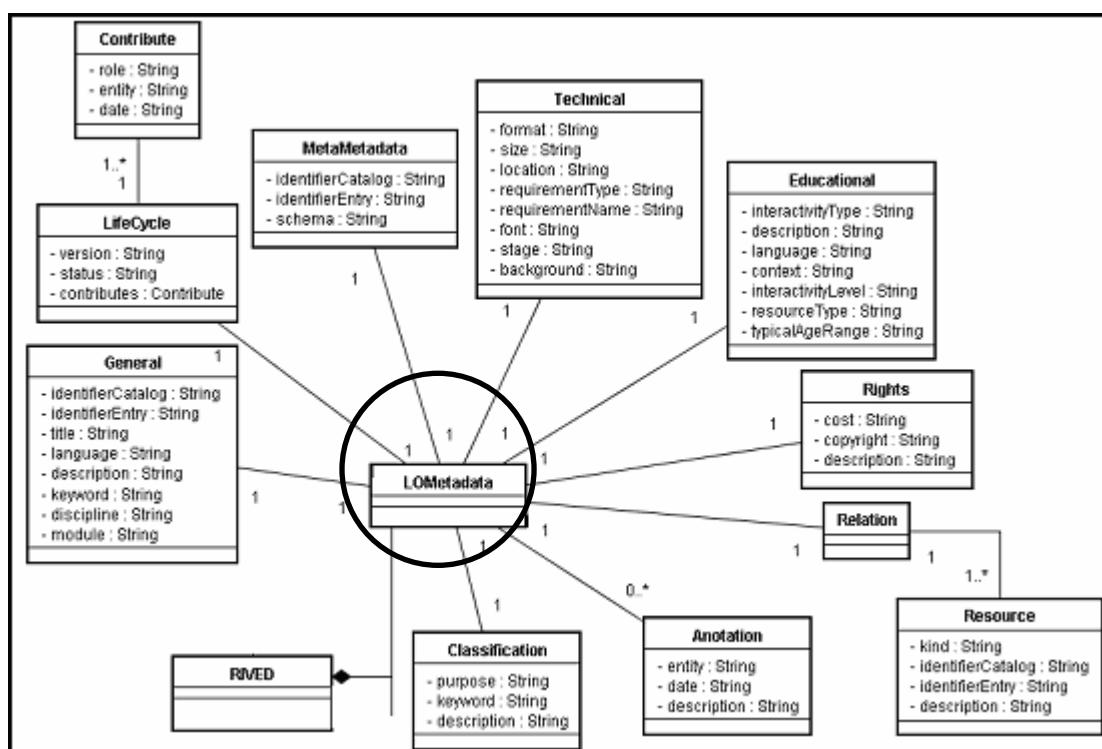


Figura 5.4. Metadados no padrão LOM da estrutura de informação proposta

Os metadados de conteúdos de um OA, conforme mostra a Figura 5.5, são representados pelas classes:

- **Cenas (Scene):** representam as situações ou contextos utilizados para apresentar os conteúdos. Por exemplo, um menino participando de uma corrida de balão, passeando no parque, uma usina emitindo gases poluentes, etc. Uma cena deve ter no mínimo 1 cenário. Utiliza-se aqui uma metáfora com o teatro, no qual a cena de uma peça é representada contendo elementos de apoio (armário, mesa, figuras) que constituem o cenário no qual se desenrola a ação em si. A **cena** então é constituída pelo conjunto de ações desenvolvidas pelos atores utilizando os recursos do cenário. No mesmo sentido, no objeto, uma cena é uma situação de contextualização mas que somente se materializa através dos elementos do cenário e das ações sobre ele.

- **Cenários (Scenary):** representam as várias telas e os diferentes elementos integrantes de cada situação em particular. Os cenários podem possuir textos, imagens, botões, tabelas e exercícios, conforme o *design* do objeto.

- **Textos (Texts):** representam todos os textos de um OA.

- **Imagens (Images):** representam todas as imagens que compõem um

OA.

- Tabelas (*Tables*): representam os conteúdos que estão em formato de Tabela.
- Botões (*Buttons*): representam todos os botões e as respectivas ações disparadas pelo acionamento dos mesmos.
- Exercícios (*Exercises*): representam os exercícios existentes em um OA. Contém questões, opções de resposta, as respostas corretas e *feedback* dado ao aluno em caso de resposta incorreta.
- Créditos (*Credits*): representam os membros da equipe, suas funções e outros colaboradores de um OA. Notar que um mesmo OA pode ter cenas desenvolvidas por pessoas diferentes, razão pela qual os créditos estão associados a Cenas e não somente ao objeto em si.

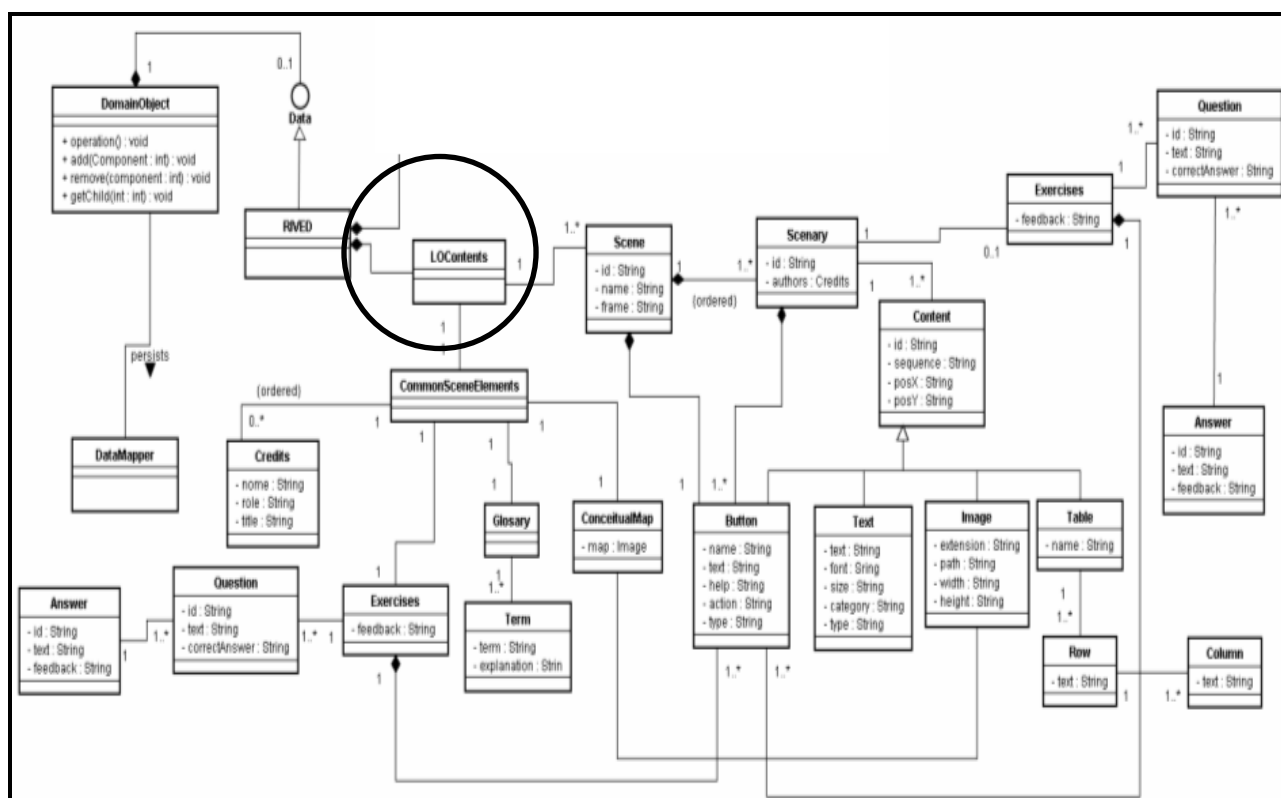








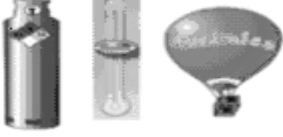





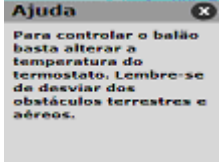

Figura 5.5. Metadados de conteúdos da estrutura de informação proposta

Além das divisões em classes ainda existem as subdivisões. O metadado de conteúdo Botão (*Button*) possui um elemento denominado tipo (*type*). Esse elemento descreve os tipos de botões, conforme mostra o Quadro 5.4.

Tipo	Descrição	Exemplos
Cenas	Permite navegar entre as cenas, não obrigando pois o usuário a seguir navegação linear.	
Navegacional	Permite navegar entre cenários de uma mesma cena.	
Exercícios	Permite acessar os exercícios gerais de um OA.	
Glossário	Permite acessar o glossário de um OA.	
Mapa Conceitual	Permite visualizar o Mapa Conceitual de um OA.	
Créditos	Permite acessar os créditos de um OA. Existem os créditos de um OA como um todo e os créditos específicos de cada cena.	
Explicativo	Permite acessar algumas informações sobre conteúdo de um OA ou algum recurso existente nele.	
Controle de animação	Permite controlar ações de objetos como, por exemplo, parar animação, continuar, etc.	
Customizado	Permite manipular diretamente um objeto como, por exemplo, movê-lo para outro lugar ou disparar uma ação como por exemplo, abrir uma garrafa (o botão seria a tampa da mesma).	
<i>Help</i>	Permite obter ajuda sobre algum recurso de um OA ou ação que pode ser realizada.	

Quadro 5.4. Tipos de botões

Da mesma forma que o metadado Botão, o metadado Texto (*Text*) possui um elemento tipo (*type*) que descreve os tipos de textos existentes, conforme mostra o Quadro 5.5.

Tipo	Descrição	Exemplos
Texto de cenário	São textos que estão presentes no cenário, independentes de qualquer ação. Servem para orientar o usuário sobre o contexto.	
Texto explicativo	São textos apresentados através de clicagem de um botão explicativo e tem como objetivo explicar algum recurso ou conteúdo.	
Texto de ajuda	São textos que aparecem quando o usuário clica no botão de <i>Help</i> . São textos que auxiliam a usar algum recurso ou como executar uma ação.	
Texto de fala de personagem	São textos que surgem em função de diálogos entre personagens eventualmente existentes no OA.	

Quadro 5.5. Tipos de Texto

A seção a seguir apresenta os recursos utilizados para a representação da estrutura de informação aqui apresentada.

5.3 Representação da estrutura de informação

Para ditar as regras que a estrutura de informação em XML deve seguir foi elaborado o XML *Schema*. A Figura 5.6 mostra parte do XML *Schema* que representa a estrutura de informação. O XML *Schema* completo encontra-se no Anexo 3. O indicador de ordem *sequence* (Figura 5.6) indica que os elementos devem seguir a ordem em que estão descritos. O indicador *complexType* (Figura 5.6) significa que um elemento é do tipo complexo, ou seja, possui outros elementos.

```

<xs:element name="LoMetadata">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Anotation"/>
      <xs:element ref="Classification"/>
      <xs:element ref="Educational"/>
      <xs:element ref="General"/>
      <xs:element ref="LifeCycle"/>
      <xs:element ref="MetaMetadata"/>
      <xs:element ref="Relation"/>
      <xs:element ref="Rights"/>
      <xs:element ref="Technical"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="Anotation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="entity"/>
      <xs:element ref="date"/>
      <xs:element ref="description"/>
    </xs:sequence>
  </xs:complexType>

```

Figura 5.6. Parte do XML *Schema* do arquivo XML da equipe RIVED-UENF

A partir do XML *Schema*, com suas restrições e especificações, pôde-se organizar a estrutura de informação em XML, a qual é responsável pela parte textual que compõe os OA.

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<object>
  <Rived>
    <LoMetadata> {Metadados LOM}
      <General>
        <identifierCatalog></identifierCatalog>
        <identifierEntry> </identifierEntry>
        <title> </title>
        <language> </language>
        <description> </description>
        <keyword> </keyword>
        <discipline> </discipline>
        <module> </module>
      </General>
      <LifeCycle> [8 lines]
      <MetaMetadata> [4 lines]
      <Technical> [8 lines]
      <Educational> [8 lines]
      <Rights> [4 lines]
      <Relation> [7 lines]
      <Anotation> [4 lines]
      <Classification> [4 lines]
    </LoMetadata>
  </Rived>
</object>

```

Figura 5.7. Parte do arquivo XML (metadados LOM) da estrutura de informação proposta

O exemplo da Figura 5.7 corresponde à parte do XML *Schema* que é apresentada na Figura 5.6. O elemento *LoMetadata* do tipo *complexType* possui vários elementos. Esses elementos devem seguir a ordem como são descritos, uma vez que isso foi definido pelo XML *Schema* através do indicador de ordem *sequence*.

A Figura 5.8 mostra parte do arquivo XML que trata as cenas de um OA. Observe que o elemento *Scene* possui um atributo *id*. Esse atributo serve como índice e seu papel é distinguir as cenas que compõem um OA. Os cenários também possuem um atributo de identificação, pelo fato de que uma cena pode possuir um ou mais cenários. Observe ainda na Figura 5.8, que dentro de um cenário existem vários elementos como, por exemplo, imagem (*Image*) e botões (*Buttons*). O arquivo XML que aparece da Figura 5.8 está reduzido com o objetivo de mostrar todos os elementos que fazem parte dos metadados de conteúdo (*LOContents*), de uma cena (*Scene*) e de um cenário (*Scenary*), por isso mostra números antes da palavra *lines* (Figura 5.8) em alguns elementos, significando que eles possuem outros elementos. Por exemplo, na frente do elemento *Text* aparece 5 *Lines* indicando que *Text* possui 5 elementos (*text*, *font*, *size*, *category* e *type*). As Figura 5.7 e Figura 5.8 mostram partes da estrutura de informação padrão em XML. A estrutura de informação completa, em XML, encontra-se no Anexo 4.

```

<LOContents> {Informações de cenas e cenários}
  <CommonSceneElements> { Informações gerais }
    <Text id="" sequence="" posX="" posY=""> [5 lines]
    <ConceitualMap> [9 lines]
    <Glosary> [13 lines]
    <Exercises> [17 lines]
    <Credits> [18 lines]
    <Button id="" sequence="" posX="" posY=""> {botoes que mudam as cenas} [6 lines]
  </CommonSceneElements>
  <Scene id="" name="" frame="">
    <Scenary id="1">
      <Content>
        <Text id="" sequence="" posX="" posY=""> [5 lines]
        <image id="" sequence="" posX="" posY=""> [5 lines]
        <Button id="" sequence="" posX="" posY=""> [6 lines]
        <Table id=""> [7 lines]
        <Exercises> [17 lines]
      </Content>
    </Scenary>
  </Scene>
</LOContents>

```

Figura 5.8. Parte do arquivo XML (que trata cenas) da estrutura de informação proposta

A seguir (Figura 5.9) é apresentada a interface do OA “Balão” após a criação da nova estrutura de informação em XML. Comparando a interface do Objeto “Balão” da Figura 3.9 com a interface da Figura 5.9, pode-se notar que no local de cenários foi colocado cenas e que a parte indicada pela seta contendo a palavra ‘Cenário’ na Figura 5.9 indica que é um cenário de uma das cenas acima, ou seja, a partir daí entende-se que uma cena possui vários cenários e esses cenários (*Scenary*) contém vários elementos, como pode ser visto na Figura 5.8.



Figura 5.9. Tela do OA "Balão" de acordo como a estrutura de informação padrão

Com o objetivo de comprovar que a nova estrutura de informação representa todos os recursos de um OA foi realizada uma prova de conceito que consistiu em substituir a estrutura de informação original do OA “Balão” pela estrutura padrão. A estrutura padrão possibilitou a descrição de todos os recursos da estrutura original de forma organizada aproveitando as potencialidades da XML, bem como, o uso dos metadados do padrão LOM permitiu incluir nessa estrutura as informações de descrição dos objetos que anteriormente eram informadas através de um formulário manual de cadastramento.

Uma cena completa do OA “Balão” descrita na estrutura padrão encontra-se no Anexo 5. A Figura 5.10 apresenta um trecho do OA “Balão” na estrutura original e a Figura 5.11 apresenta o mesmo trecho na estrutura padrão.

```

<!-- Textos em camadas-->
<screen index="2">
  <texts>
    <text>Clique nos cilindros para ver as características dos gases contidos nos mesmos.</text>
    <text>Neste Parque você pode conhecer a casa dos gases e os mistérios de alguns balões; pode ir a
    <text>Agora você pode verificar o comportamento dos gases arrastando até três cilindros cada vez
    <text>Passe o mouse sobre o balão para ver o gás em seu interior.</text>
  </texts>
</screen>

```

Figura 5.10. Trecho do OA “Balão” na estrutura original

```

<Scenary id="9">
  <Content>
    <Text id="1" sequence="" posX="" posY="">
      <text>Clique nos cilindros para ver as características dos gases
      <u>contidos nos mesmos.</u></text>
      <font>Arial </font>
      <size> 16 </size>
      <category>explicativo</category>
    </Text>

```

Figura 5.11. Trecho do OA “Balão” na estrutura padrão

A Figura 5.12 apresenta o trecho inicial da estrutura de informação original (Figura 5.12a) e da estrutura padronizada (Figura 5.12b). Observe que o início da estrutura original começa descrevendo as telas (equivalente a cenários na estrutura atual) enquanto a estrutura atual começa descrevendo os metadados do padrão LOM.

<pre> <?xml version="1.0" encoding="iso-8859-1"?> <explanation> <screens> <!-- Textos da padronização --> <screen index="0"> <texts> <text>Química</text> <text>Interface 0</text> <text>RIVED</text> </texts> </screen> <!-- Apresentação/Créditos --> <screen index="1"> <texts> <text>Casa dos</text> <text>Gases</text> <text>Clevis Elena Rapkiewicz, D.Sc.</text> <text>coordenação tecnológica</text> <text>Maria Cristina Canela, D.Sc.</text> <text>coordenação pedagógica</text> <text>Eduardo Coelho Carneiro</text> <text>desenvolvimento</text> <text>Gláucia Ribeiro Gonzaga</text> <text>design gráfico e modelagem</text> <text>Valéria Machado da Costa</text> <text>avaliação visual e supervisão</text> <text>ENTRAR</text> </texts> </screen> <!-- Textos em camadas--> <screen index="2"> <texts> <text>Clique nos cilindros para ver as características </pre> <p style="text-align: center;">Figura A</p>	<pre> <?xml version="1.0" encoding="iso-8859-1" standalone="no"?> <object> <Rived> <!Metadata> <General> <identifierCatalog>1</identifierCatalog> <identifierEntry>0A_Balão</identifierEntry> <title>Casa dos Gases</title> <language>Português</language> <description> O objeto Balão explica diferentes tipos de gases.</description> <keyword> gases, temperatura, balão</keyword> <discipline> Química</discipline> <module> modulo 1</module> </General> <LifeCycle> <version>final </version> <status>pronto </status> <contributors> <role>Universidade</role> <entity> UENF</entity> <date>08/09/2008</date> </contributors> </LifeCycle> <MetaMetadata> <identifierCatalog> 01 </identifierCatalog> <identifierEntry> 0ABalao </identifierEntry> <schema>novo_xml.xsd </schema> </MetaMetadata> <Technical> <format> swf </format> <size> 2.991 Kb </size> <location> </location> <requirementType>plugin</requirementType> <requirementName>plugin Flash</requirementName> </pre> <p style="text-align: center;">Figura B</p>
---	---

Figura 5.12. OA “Balão” - Início da Estrutura de informação original do OA (a) e da Estrutura de informação padrão (b) (com metadados do padrão LOM)

Na estrutura antiga não havia os conceitos de cenas e cenários, conforme apresenta a Figura 5.12a. Já a estrutura atual de um OA (Figura 5.12b) possui várias cenas e essas possuem cenários que, por sua vez possuem imagens, textos, botões, etc. Uma outra mudança foi em relação ao uso das informações de forma a permitir maior flexibilidade como, por exemplo, na estrutura antiga os botões eram descritos de forma geral para todas as telas de um OA e na estrutura atual cada cenário possui uma descrição de botões, imagens, textos, etc., conforme mostra a Figura 5.13b.

<pre> <!-- Textos em cenas --> <screen index="3"> <text> <text>Clique nos cilindros para ver as características dos gases contidos nos mesmos.</text> <text>Neste Parque você pode conhecer a casa dos gases e os mistérios de alguns balões...</text> <text>Agora você pode verificar o comportamento dos gases arrastando até três cilindros...</text> <text>Resse o mouse sobre o balão para ver o gás em seu interior.</text> </text> </screen> <!-- Botões --> <screen index="4"> <text> <text>Limpar</text> <text>Testar</text> <text>Continuar</text> <text>Reiniciar</text> <text>Como navegar</text> </text> </screen> <!-- HELPS DE CONTEXTO --> <screen index="5"> <text> <text>Entrar no objeto</text> <text>Retornar</text> <text>Avançar</text> <text>Iniciar</text> <text>Retornar ao início do objeto</text> <text>Créditos</text> <text>Ocultar explicações</text> <text>Ajuda</text> <text>Fechar a ajuda</text> </pre>	<pre> <Scene id="" name="Inicial" frame="" (1ª CENA- INICIAL) <Screen id="1"> <Content> <Text id="1" sequence="1" posX="" posY=""> <text>Neste menu você encontra os links para o início do módulo, os exercícios de fixação, o mapa de conceitos do objeto, o glossário e os créditos.</text> Arial <size> 12 </size> <category> explicativo </category> </Text> <Text id="2" sequence="2" posX="" posY=""> <text>Este é o menu de navegação de cenários. Cenários são as diferentes situações que você vai encontrar no objeto. Os números indicam a quantidade de cenários. Aqui você pode escolher por onde quer começar a sua navegação.</text> Arial <size> 12 </size> <category> explicativo</category> </Text> <Text id="3" sequence="3" posX="" posY=""> <text>Aqui ficam os botões de navegação pelas telas. Utilize a seta para a direita para avançar e a seta para esquerda para retornar à tela anterior. O botão central leva à tela inicial.</text> Arial <size> 12 </size> <category> explicativo</category> </Text> <Button id="1" sequence="" posX="" posY=""> <name>Entrar </name> <text> Entrar</text> <help>Entrar no Objeto</help> <action>Iniciar o Objeto</action> <type>navegacao</type>/> </Button> <Button id="2" sequence="" posX="" posY=""> <name>Como Navegar </name> <text> Como navegar</text> </pre>
Figura A	Figura B

Figura 5.13. OA “Balão” - Estrutura de informação original do OA (a) e a Estrutura de informação padrão (b) que trata cenas do OA

Enfim, após a substituição da estrutura de informação original pela estrutura padrão pôde-se observar que essa última descreve todos os recursos de um OA e ainda permite uma descrição de metadados do padrão LOM, permitindo uma descrição mais detalhada dos recursos e uma melhor utilização dos OA. Essa estrutura atual também é mais abrangente, pois foi elaborada através da pesquisa realizada com outros OA. Além disso, a possibilidade de cada cena e cenário possuir um conjunto de elementos próprios, como pôde ser visto na Figura 5.13b, permite que a exclusão de cenário ou de uma cena seja feita sem que isso prejudique a parte textual de um OA, permitindo maior flexibilidade e reusabilidade. Com a padronização da estrutura de informação realiza-se a reusabilidade de conteúdos como, por exemplo, um OA poder usar a cena ou o cenário de outro OA. Por exemplo, se um OA possui um cenário que trata da poluição atmosférica, esse mesmo cenário pode ser utilizado por outro OA em um contexto diferente.

Uma vez que tenha sido feita a padronização e a representação da estrutura, a etapa seguinte foi o desenvolvimento de uma ferramenta de fácil utilização pela equipe de domínio permitindo a criação e edição da parte textual dos OA, o que é apresentado no próximo capítulo.

6. EDITANDO A PARTE TEXTUAL DOS OBJETOS E OS METADADOS

Conforme destacado no capítulo referente à metodologia, foram analisados editores genéricos para XML, sendo que nenhum atendeu aos requisitos considerados necessários para ser usado pela equipe da área de domínio no processo de desenvolvimento de OA. Por outro lado, a análise dos mesmos (seção 6.1) propiciou estabelecer os casos de uso para o desenvolvimento de um editor específico. A seção 6.2 apresenta os casos de uso estabelecidos para a o editor. A seção 6.3 descreve a plataforma tecnológica utilizada no desenvolvimento. Finalmente, a seção 6.4 apresenta o protótipo desenvolvido e a avaliação do mesmo.

6.1 Análise dos Editores

Os editores analisados foram o *Oxygen*, *Altova* e o *XML Notepad*, sendo o último gratuito.

O *XML Notepad* apresenta em seu lado esquerdo (Figura 6.1), os elementos do arquivo (*i.e.*, as *tags* ou estrutura XML), em formato de árvore. Ao clicar-se sobre os elementos, no lado esquerdo é apresentado o texto correspondente ao elemento (o conteúdo), permitindo sua alteração, sem controle nem restrições.

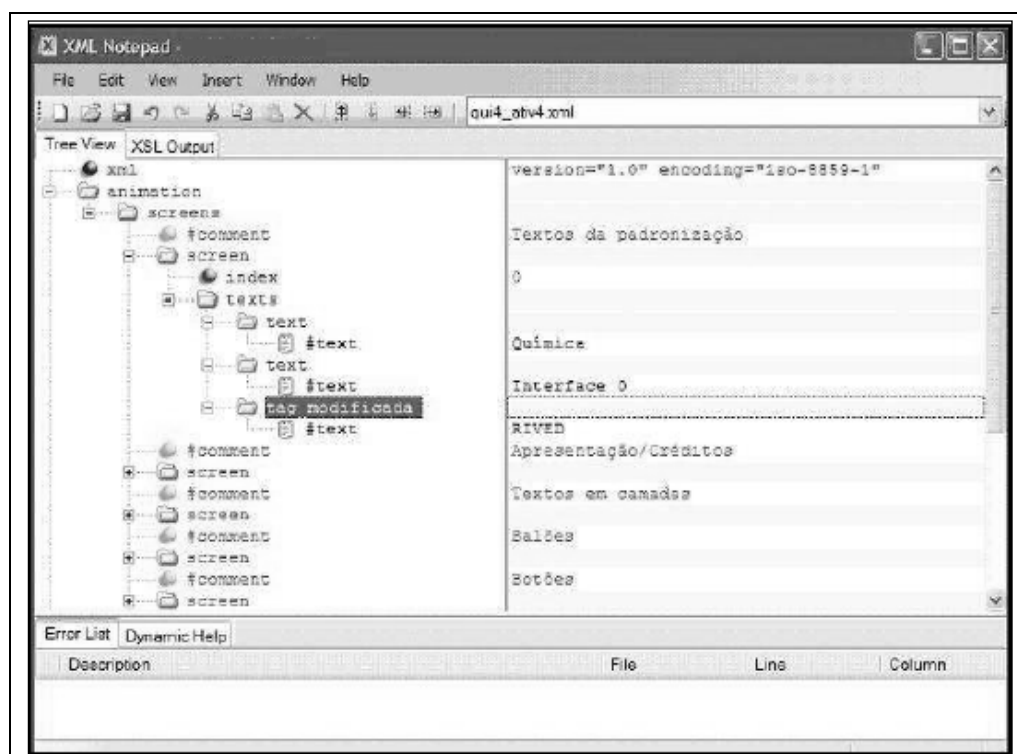


Figura 6.1. Interface no XML Notepad

O editor *Oxygen* permite criar e editar documentos como XML, XSL, TXT e DTD. Neste editor é possível abrir o arquivo XML no formato *Text* (Figura 6.2a) e no formato *Grid* (Figura 6.2b).

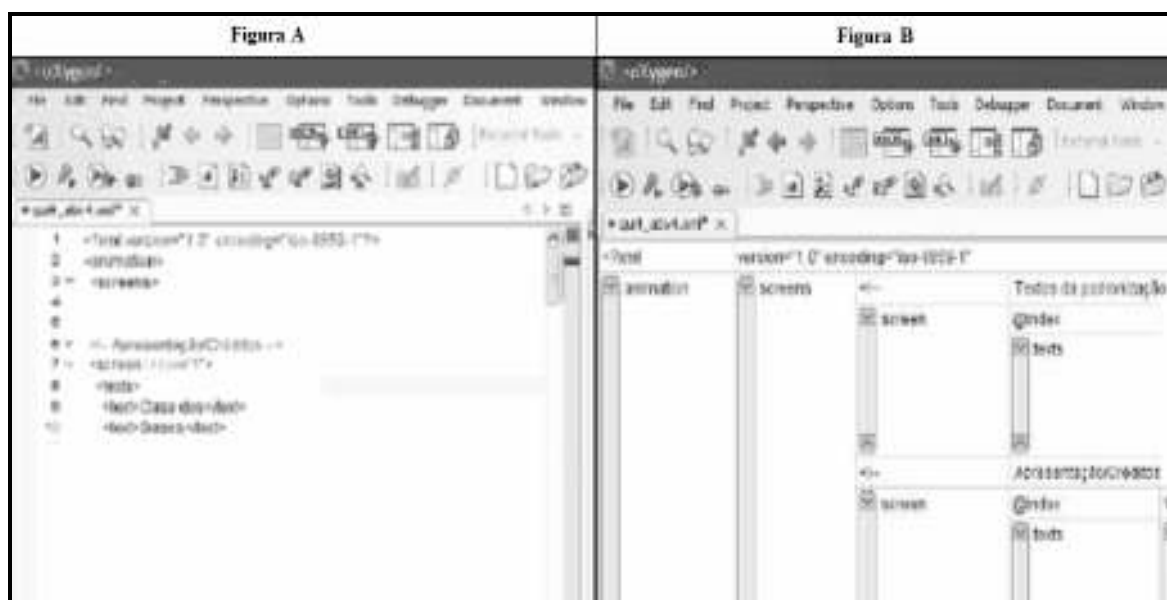


Figura 6.2. Modelo *Text* e Modelo *Grid* do *Oxygen*

Similar ao *Oxygen*, o *Altova* apresenta o conteúdo na opção *Text* e também na opção *Grid*, conforme mostra a Figura 6.3.

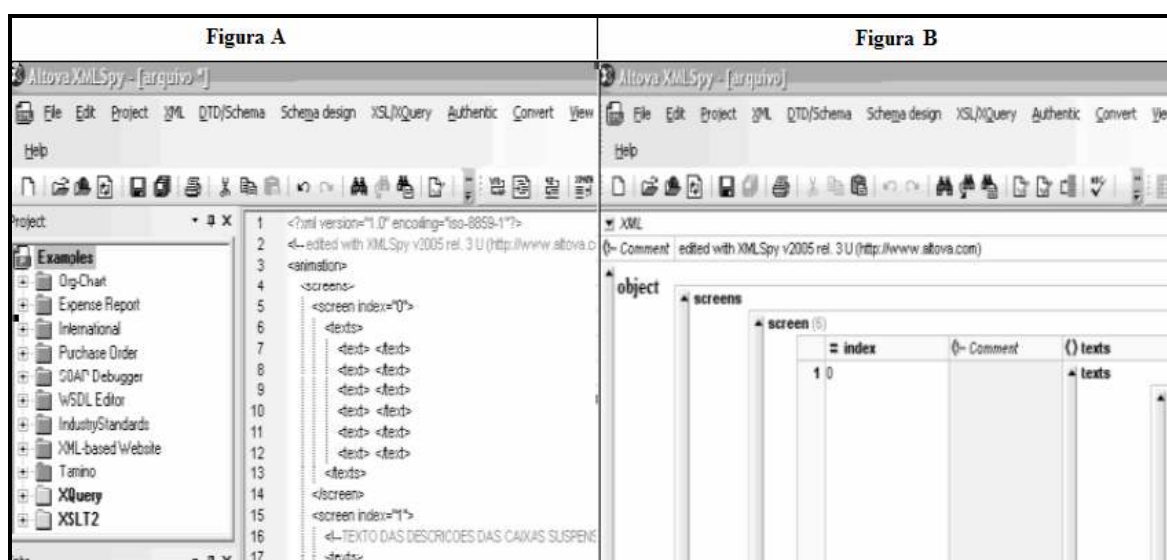


Figura 6.3. Modelo *Text* (a) e Modelo *Grid* (b) do *Altova*

Os editores foram utilizados por um professor de Química do CEFET-Campos, o qual interage com a equipe RIVED-UENF desde 2006. Para análise dos editores foi desenvolvida toda a parte textual de um OA ainda não existente, uma

vez que o intuito foi analisar a viabilidade de passar a edição da parte textual dos OA para a equipe de domínio. Uma mesma cena de objeto foi editada nos três editores citados. Conforme já citado na metodologia, após a edição, o referido professor preencheu um questionário (Anexo 6) que teve como objetivo verificar se os editores são adequados para editar a estrutura de informação dos OA e fez um relato escrito analisando o processo de uso de cada editor. O Quadro 6.1 apresenta um comparativo das respostas do professor ao analisar cada um dos editores.

EDITORES XML	O X Y G E N	X M L N O T E P A D	A L T O V A
A - Critérios baseados na ISO 9126 – Funcionalidade			
A1- O <i>software</i> faz o que é apropriado.	S	S	S
A2- O <i>software</i> faz de forma correta o que foi proposto.	P	P+	P
B - Critérios baseados na ISO 9126 – Usabilidade			
B1- O <i>software</i> é de fácil aprendizado.	P	P-	P
B2- As representações das funções de interface (botões, ícones, <i>helps</i> , etc.) são fáceis de serem entendidas.	P	P	P-
B3- O editor é autoajustável de acordo com o tamanho do monitor, permitindo boa visibilidade.	S	S	S
B4- Os <i>helps</i> dos botões (ativados quando o cursor do mouse está sobre os botões) auxiliam quanto ao uso.	P	P	P
B5- O sistema de ajuda é adequado, ou seja, explica as dúvidas do usuário adequadamente.	P+	P+	P+
B6- Os títulos dos menus são satisfatoriamente explicativos, refletindo a natureza da escolha a ser feita.	P	P	P
B7- Os textos existentes entre cada <i>tag</i> de abertura e fechamento aparecem de forma organizada.	P	S	S
B8- O <i>software</i> é de fácil operação e controle.	P-	P	P-

C - Critérios baseados na elicitação de requisitos para um editor de OA			
C1- O <i>software</i> exibe as <i>tags</i> de abertura e fechamento (<> </>).	S	P	S
C2- O <i>software</i> permite renomear o conteúdo das <i>tags</i> .	S	P	S
C3- É complexo criar e alterar o arquivo XML com o <i>software</i> .	P	P	P
C4- O <i>software</i> oferece/permite uma padronização (utiliza um padrão) para a estrutura de informação dos OA.	P	S	P
C5- O <i>software</i> permite inserir <i>tags</i> e seu conteúdo .	S	P	S
C6- O <i>software</i> permite remover <i>tags</i> e seu conteúdo.	S	P	S
C7- O <i>software</i> permite a validação do arquivo XML.	S	S	S
C8- O <i>software</i> parece adequado para a criação de arquivos XML para OA.	P	P+	P

Quadro 6.1. Análise dos editores

Segue abaixo o relato do professor sobre o uso do editor *Oxygen*:

[...] acredito que, principalmente nesses cenários [Figura 6.4] cometi inúmeros erros no momento de adicionar as *tags*, os textos, as imagens e os botões. Mesmo porque esse editor XML permite que as *tags* sejam apagadas por um equívoco do programador.

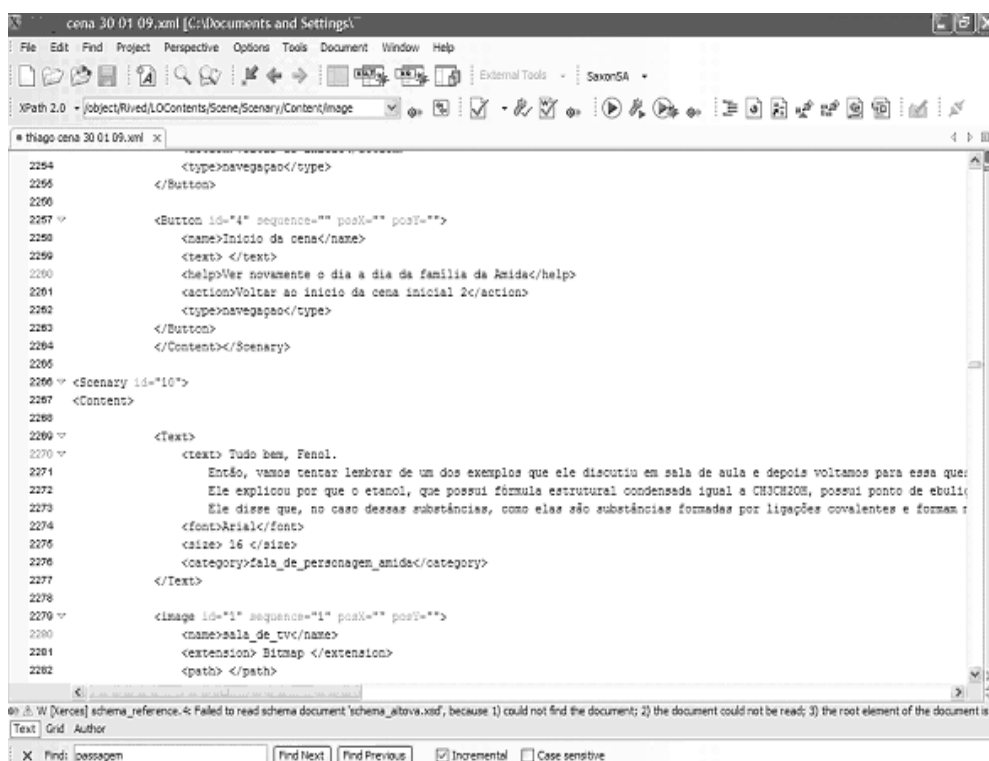


Figura 6.4. Exemplo de uso do *Oxygen*

O relato do professor ao utilizar o editor *Altova*:

[...] O editor *Altova* é bem similar ao *Oxygen*, sendo assim, as dificuldades encontradas para realizar a programação em XML nesse editor foram análogas às encontradas no editor *Altova*. Assim como no *Oxygen*, neste editor, o programador pode apagar uma determinada *tag*, de fechamento, por exemplo, sem ser alertado automaticamente pelo *software*. Dessa forma, ao final da programação de um determinado objeto vários erros podem ter sido cometidos pelo programador sem que ele tenha percebido. Com relação a algumas diferenças existentes entre estes *softwares*, vale ressaltar apenas que, no caso do *Altova*, a identificação do início e do final de um determinado elemento, texto, por exemplo, é facilitada. Isso ocorre, por que existe uma linha que liga a *tag* de início com a *tag* de fechamento no próprio texto da programação [Figura 6.5 detalhe A]. No caso do *Oxygen*, essa

verificação só é feita quando o cursor é colocado no canto esquerdo da tela.

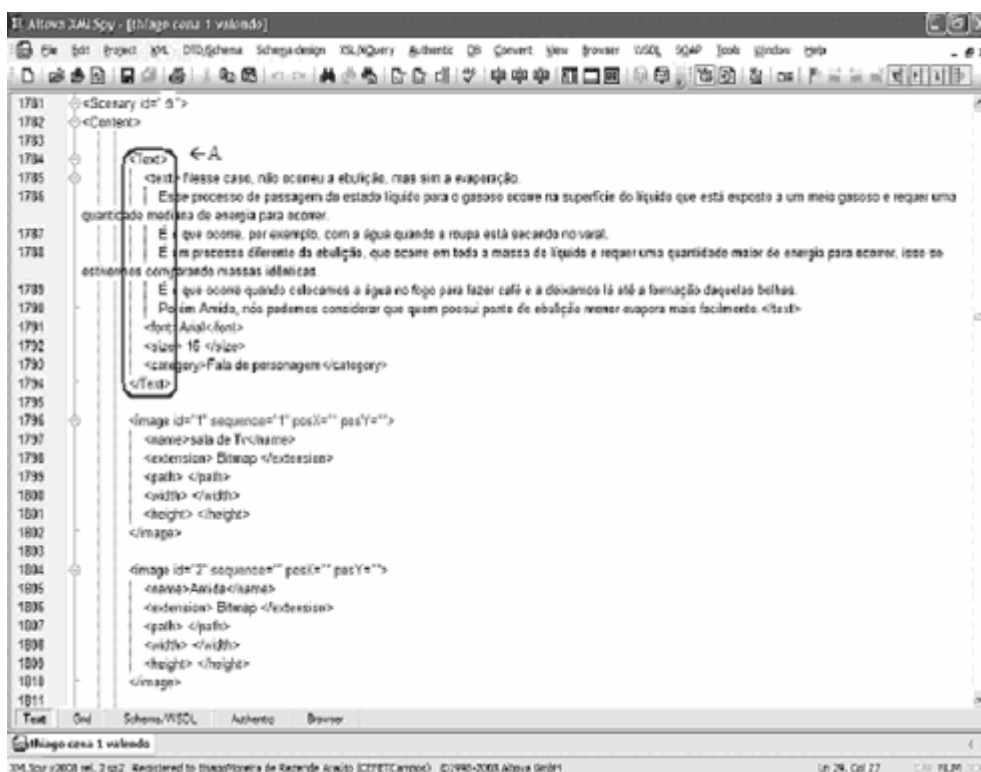


Figura 6.5. Exemplo de uso do *Altova*

O relato feito pelo professor ao utilizar o editor XML *Notepad*:

[...] O XML *Notepad* possui interface um pouco diferente dos demais *softwares* utilizados para proceder à programação do objeto [termo utilizado pelo professor referindo-se a edição da parte textual de um OA. O mesmo chamou de *programação* por visualizar as *tags*]. Ele apresenta os elementos, em formato árvore, do lado esquerdo e os textos correspondentes do lado direito [Figura 6.6]. Essa forma de apresentação faz com que fique mais difícil que o programador cometa erros, tal como: apagar uma *tag* de fechamento equivocadamente, já que ele só poderá alterar a *tag* se der um clique duplo com o mouse em um dos elementos do lado esquerdo. Ou seja, o acesso as *tags* não é tão livre como nos demais *softwares*. Esse esquema de apresentação auxiliou na programação do objeto.

E ainda sobre o XML *Notepad*:

[...] esse tipo de apresentação em árvore dificulta a visão geral da programação, o que não ocorre com os demais *softwares* utilizados, o que poderia se tornar um empecilho para a realização da programação. Porém, o *software* XML *Notepad* também permite essa visão geral, bastando para isso utilizar a opção de *XSL output*.

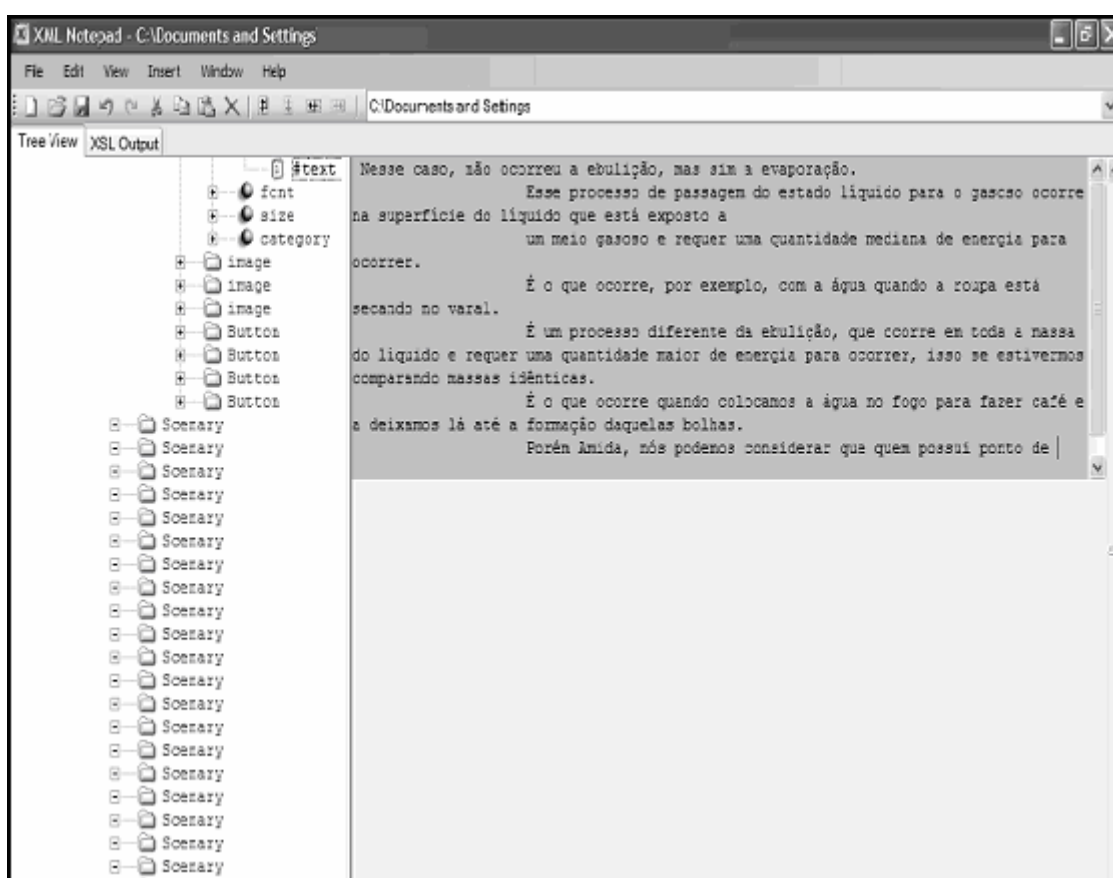


Figura 6.6. Exemplo de uso do XML *Notepad*

De acordo com a avaliação, os editores apresentam pontos positivos e negativos. Entre os pontos positivos do *Oxygen*, o editor é autoajustável, possui sistema de ajuda adequado e permite validação de um documento XML. Entre os pontos negativos: as *tags* são exibidas, editáveis, permite inserir e remover *tags* e conteúdos. Já o *Altova*, apresenta os mesmos pontos positivos do *Oxygen* e ainda, os textos entre *tags* de abertura e fechamento aparecem de forma organizada. Mas

também permite que as *tags* sejam exibidas, editadas e permite inserção e remoção de *tags* e conteúdos.

Apesar de existirem editores gratuitos e proprietários que permitem editar arquivos XML, tais como o XML *Notepad*, o *Oxygen* e o *Altova*, eles não são adequados para a equipe em questão como apresentado no Quadro 6.2.

Problemas encontrados nos editores	
São genéricos	Permitem abrir qualquer documento XML, inclusive documentos que não estão bem formados.
Permitem modificar o conteúdo do elemento	O usuário pode modificar o nome de um elemento, o que causaria erro na exibição de um OA.
Não utilizam um padrão de metadados	Não existe um padrão de metadados como, por exemplo, o LOM para auxiliar na descrição de um OA.
Permitem incluir novos elementos e removê-los	O usuário pode inserir e remover elementos que não estão presentes na estrutura padrão ou excluir um elemento que é obrigatório, o que causaria erros na exibição de um OA.

Quadro 6.2. Problemas comuns nos editores analisados

Em virtude dos problemas, apresentados no Quadro 6.2, nos editores analisados, o passo seguinte foi o estudo das tecnologias para o desenvolvimento da ferramenta específica para o processo apresentado.

De acordo com o bloco B do Quadro 6.1, chegou-se a conclusão de que a ferramenta deve possuir interface amigável e intuitiva, uma vez que as questões referentes a facilidade de aprendizado e de operação e controle tiveram resultados entre P (Parcialmente) e P- (Parcialmente com muitas restrições). É fundamental que os aspectos técnicos da XML, como o uso de *tags*, sejam imperceptíveis ao usuário, no caso, uma equipe pedagógica da área de domínio.

6.2 Casos de uso

Após verificar que os editores não são apropriados para a edição da estrutura de informação, o passo seguinte foi iniciar a modelagem do sistema

utilizando os Casos de Uso da UML para explicitar os requisitos para o desenvolvimento da ferramenta. Os Casos de Uso são apresentados na Figura 6.7.

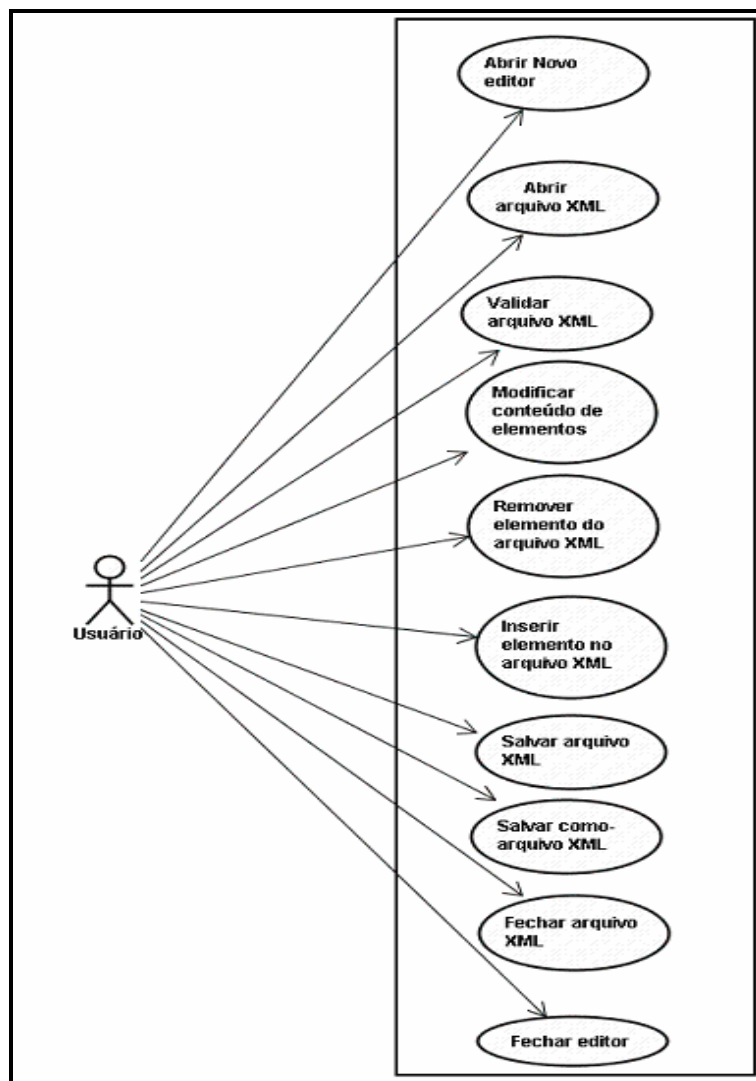


Figura 6.7 . Casos de uso do editor

O Quadro 6.3 apresenta uma explicação dos casos de uso, visando tornar clara a função de cada um deles.

Casos de uso	O que permite
Abrir arquivo novo	Iniciar um novo arquivo XML quando não há semelhança do objeto em desenvolvimento com nenhum outro já desenvolvido pela equipe.
Abrir arquivo XML	Abrir um arquivo já existente para consultar a parte textual do objeto ou fazer alterações sobre a mesma.
Modificar conteúdo dos elementos	Modificar conteúdos sem modificar a estrutura (ex. mudar o texto de um <i>help</i> ou de um botão).
Remover elementos	Remover elementos, por exemplo, um <i>help</i> que deixe de ser necessário ou até mesmo uma cena inteira, permitindo assim que parte do objeto não seja utilizada pelo professor, facilitando a reusabilidade.
Inserir elementos	Inserir elementos na estrutura XML. Por exemplo, inserir um <i>help</i> onde não existia.
Validar arquivo XML	Verificar se o arquivo está de acordo com a padronização proposta.
Salvar arquivo	Encerrar as ações sobre um determinado arquivo.
Salvar como	Criar outra versão de um arquivo XML e operar sobre o mesmo, inserindo elementos, modificando, etc. Ou seja, facilita criar a estrutura de novos OA que tenham partes em comum com OA já desenvolvidos.
Fechar arquivo XML	Encerrar ações sobre um determinado arquivo.
Fechar editor	Encerrar o uso do editor.

Quadro 6.3. Casos de uso

Fonte: adaptado de Santos, Rapkiewicz e Wives (2008b)

A seguir encontram-se os cenários de cada um dos Casos de Uso.

Abrir arquivo XML

Cenário Principal de Sucesso:

1. O usuário abre o editor.

2. O usuário clica no menu abrir ou no ícone que representa a ação abrir.
3. O usuário busca o arquivo através da janela que aparece na tela.
4. O usuário clica no nome do arquivo e na opção abrir.
5. O arquivo aparece na tela do editor.

Extensões:

4a: Arquivo Válido

- 1: Se o arquivo for válido então, vai para o passo 5.

4b: Arquivo não válido

- 1: Se o arquivo não for válido, o arquivo não é aberto.

Validar arquivo XML

Cenário Principal de Sucesso:

1. O usuário tenta abrir um arquivo XML.
2. O usuário escolhe o arquivo a ser aberto.
3. O sistema verifica se o arquivo escolhido está válido.

Extensões:

3a: Verificação do arquivo. Essa verificação é feita comparando o arquivo XML ao XML *Schema*.

- 1: Se arquivo for válido então, o arquivo é aberto.

2: Se o arquivo não for válido, o arquivo não é aberto. Aparece uma mensagem informando que o arquivo não é válido.

Modificar conteúdo do Arquivo XML

Cenário Principal de Sucesso:

1. O usuário após abrir o arquivo procura o conteúdo do elemento a ser modificado no formulário.
2. O usuário modifica o conteúdo.

Extensões:

2a: Elemento vazio

- 1: Se o elemento é vazio (sem conteúdo), ele é removido.

2b: Elemento obrigatório

1: Caso seja obrigatório o preenchimento do conteúdo do elemento, aparece uma mensagem indicando erro, caso não seja preenchido. O elemento é

obrigatório se o XML *Schema* defini-lo como *required*, ou seja, requerido. Essa verificação é feita através do XML *Schema*.

Inserir elemento no Arquivo XML

Cenário Principal de Sucesso:

1. O usuário clica no botão “+ nome do elemento”.
2. O elemento aparece na árvore na parte esquerda da tela.
3. Ao clicar sobre o elemento inserido aparece um formulário para preenchimento no lado direito da tela.

Extensões:

3a: Elemento vazio

- 1: Se o elemento é vazio (sem conteúdo) não aparece no arquivo XML.

3b: Elemento obrigatório

1: Caso seja obrigatório o preenchimento do conteúdo do elemento, aparece uma mensagem indicando erro, caso não seja preenchido. Essa verificação é feita através do XML *Schema*.

Remover elemento do Arquivo XML

Cenário Principal de Sucesso:

1. O usuário clica no elemento a ser removido.
2. O usuário clica no botão “- nome do elemento”.
3. O elemento é removido da árvore.

Extensões:

2a: Remoção de elemento

1: Caso seja obrigatória a existência do elemento, aparece uma mensagem indicando erro e não permitindo a remoção. Essa verificação é feita através do XML *Schema*.

Salvar arquivo XML

Cenário Principal de Sucesso:

1. O usuário clica no menu salvar ou no ícone que representa ação salvar.
2. O usuário escolhe o caminho onde deseja salvar o arquivo.
3. O usuário digita o nome que deseja salvar o arquivo.
4. O usuário clica em salvar.

5. O arquivo é salvo.

Extensões:

1a: Arquivo ainda não foi salvo

1: Se o arquivo ainda não estiver sido salvo vai para passo 2.

1b: Arquivo já foi salvo

1: Se o arquivo já estiver sido salvo vai para o passo 5.

Salvar como – Arquivo XML

Cenário Principal de Sucesso:

1. O usuário clica no menu salvar como.

2. O usuário escolhe o caminho onde deseja salvar o arquivo.

3. O usuário digita o nome que deseja salvar o arquivo.

4. O usuário clica em salvar.

5. O arquivo é salvo.

Extensões:

4a: Arquivo existente

1: Se existir um arquivo com o mesmo nome surge uma mensagem perguntando se o usuário deseja substituir o arquivo ou não. Se o usuário escolhe “não” ele pode colocar outro nome diferente do existente e salvar o arquivo. Se escolher “sim” o arquivo é substituído.

Abrir Novo Editor

Cenário Principal de Sucesso:

1. O usuário clica no menu Novo ou no ícone que representa ação de abrir Novo.

2. Aparece um novo editor.

Extensões:

1a: Arquivo não foi salvo

1: Se o arquivo em uso ainda não foi salvo, surge uma janela com a opção de salvar o arquivo.

Fechar Editor

Cenário Principal de Sucesso:

1. O usuário clica no menu Arquivo/ Fechar ou no botão que representa ação de Fechar.
2. A área de trabalho do editor é fechada.

Extensões:

1a: Arquivo não foi salvo

1:Se o arquivo em uso ainda não foi salvo, surge uma janela com a opção de salvar o arquivo.

Sair do editor

Cenário Principal de Sucesso:

1. O usuário clica no menu Sair ou no ícone que representa a ação de Sair.
2. O editor é fechado.

Extensões:

1a: Arquivo não foi salvo

1:Se o arquivo em uso ainda não foi salvo, surge uma janela com a opção de salvar o arquivo.

6.3 Plataforma tecnológica para criação da ferramenta

A linguagem escolhida para a criação da ferramenta foi Java devido às suas características, tais como: orientada a objetos (OO), independente de plataforma e gratuita. Juntamente com a linguagem Java, fez-se necessário o uso de uma API (*Application Programming Interface*) para permitir a interpretação da XML de forma fácil e satisfatória para programas de computador. Dentre as API existentes que foram analisadas, destacam-se a *Simple API for XML (SAX)*, o *Document Object Model (DOM)* e a *Java Document Object Model (JDOM)*. Através do DOM é possível acessar as informações contidas nos documentos XML. DOM é um conjunto de interfaces que decompõe um documento XML em uma árvore transversal hierárquica de nós/objetos (VELOSO, 2007). O DOM é uma interface de programação especificada pela W3C para o tratamento e manipulação de dados em documentos XML. Aborda documentos XML como uma árvore de nós. O SAX é um conjunto de interfaces que decompõe um documento XML em uma sucessão linear

de chamadas de métodos (VELOSO, 2007). Algumas características do DOM e do SAX são encontradas em Veloso (2007):

- DOM é uma solução natural para lidar com dados que são mais bem descritos como uma árvore;
- É mais conveniente usar SAX em dados estruturados ou gerados pelo computador;
- DOM agiliza a pesquisa por elementos, bastando apenas indicar o caminho dos nós na árvore;
- SAX utiliza menos memória que o DOM, pois não precisa manipular uma estrutura de dados complexa, como uma árvore em memória;
- SAX decompõe um documento XML em uma sucessão linear de chamadas de métodos, enquanto o DOM decompõe em uma árvore hierárquica de objetos.

Ao contrário do SAX, o DOM processa o arquivo XML de uma só vez e o conteúdo é representado em forma de árvore (Figura 6.8).

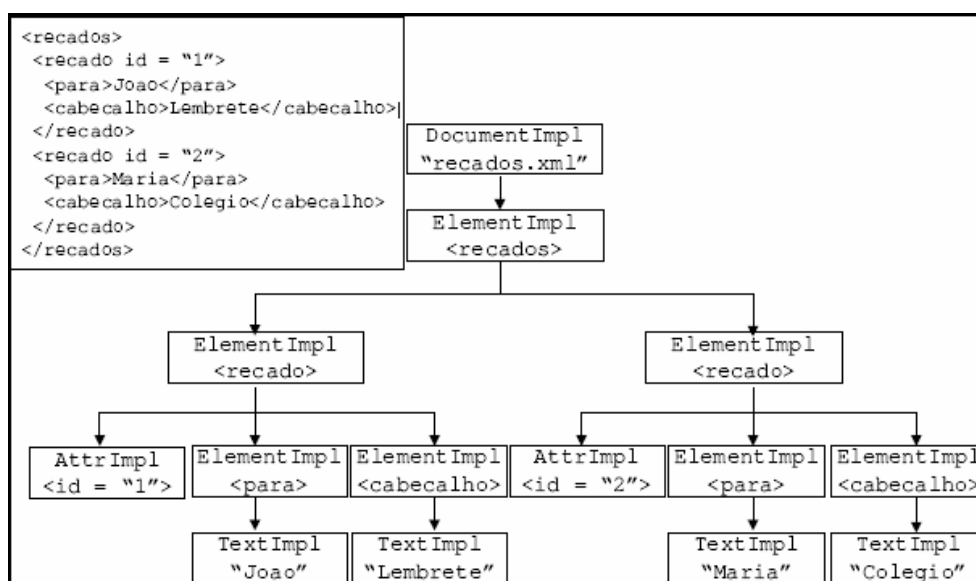


Figura 6.8. Exemplo de um documento XML no formato DOM

Fonte: Vieira, Ruberg e Mattoso (2002)

Para armazenar a árvore é preciso utilizar mais memória. A API DOM define a estrutura lógica de um documento e a forma como é acessado e manipulado (W3C, 2000). Dessa forma, é possível ao usuário manipular as informações que

constituem o conteúdo do documento, como elementos, atributos, comentários, entre outros.

O JDOM é uma API Java baseada em árvore, de código-fonte aberto para transformação, criação, manipulação e serialização de documentos XML (HAROLD, 2003). JDOM é mais abrangente do que SAX e, diferente do DOM, permite análise gramatical de documentos. A Figura 6.9 apresenta um exemplo de uso do JDOM.

```
Element element = new Element("Teste");  
element.setText("olá");  
element.setAttribute("index", "1");
```

Figura 6.9. Exemplo de código JDOM

De acordo com Nascimento e Silva (2007), o JDOM é um modelo de documento Java específico que interage com XML, mais simples e mais rápido do que a implementação DOM. De fato, após a análise das API, foi utilizada JDOM por ser fácil e simples de implementar.

A implementação foi realizada utilizando o *Eclipse* (Figura 6.10). O *Eclipse* é uma plataforma de desenvolvimento livre, extensível e muito utilizada para o desenvolvimento em plataforma Java.

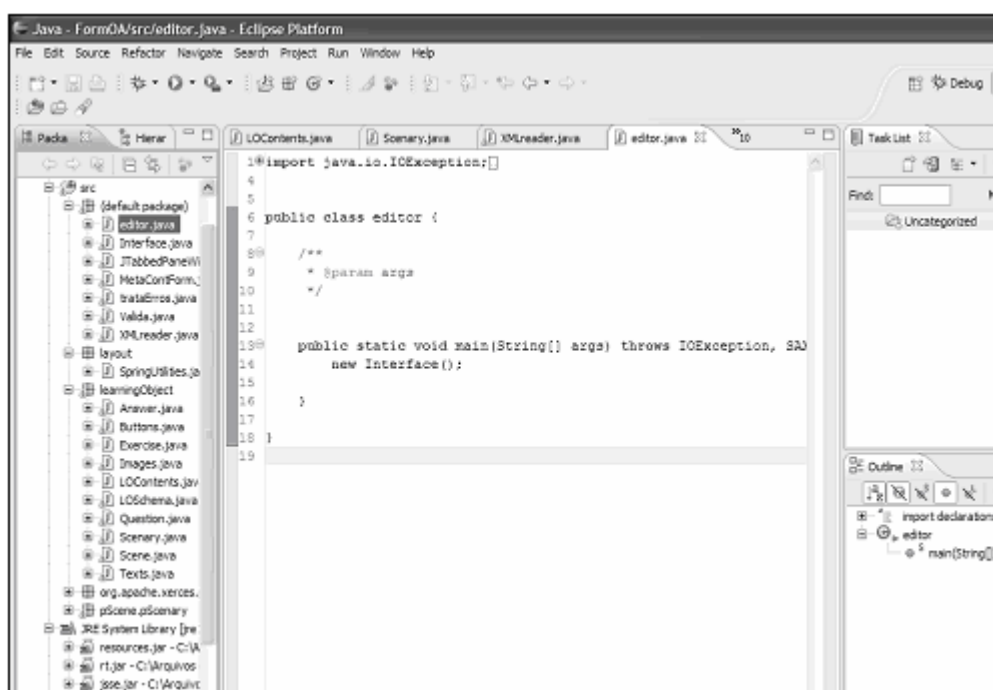


Figura 6.10. Interface da Plataforma *Eclipse*

A ferramenta permite editar os metadados LOM e os conteúdos de um OA, conforme a estrutura de informação padrão.

Concluída a implementação da ferramenta foi realizada a avaliação (seção 6.4) para verificar se o uso da ferramenta resolve os problemas descritos no capítulo 3.

6.4 Avaliação da ferramenta

Após a criação do protótipo da ferramenta, a etapa seguinte foi a de avaliação. A avaliação da ferramenta foi feita pelo mesmo professor que analisou os editores. Para avaliar a ferramenta, o professor editou uma cena do OA que ele está desenvolvendo. Após a edição, ele relatou sua experiência ao usar a ferramenta e, por fim, respondeu ao questionário.

A Figura 6.11 apresenta a interface da ferramenta desenvolvida que permite editar os metadados LOM de um OA.

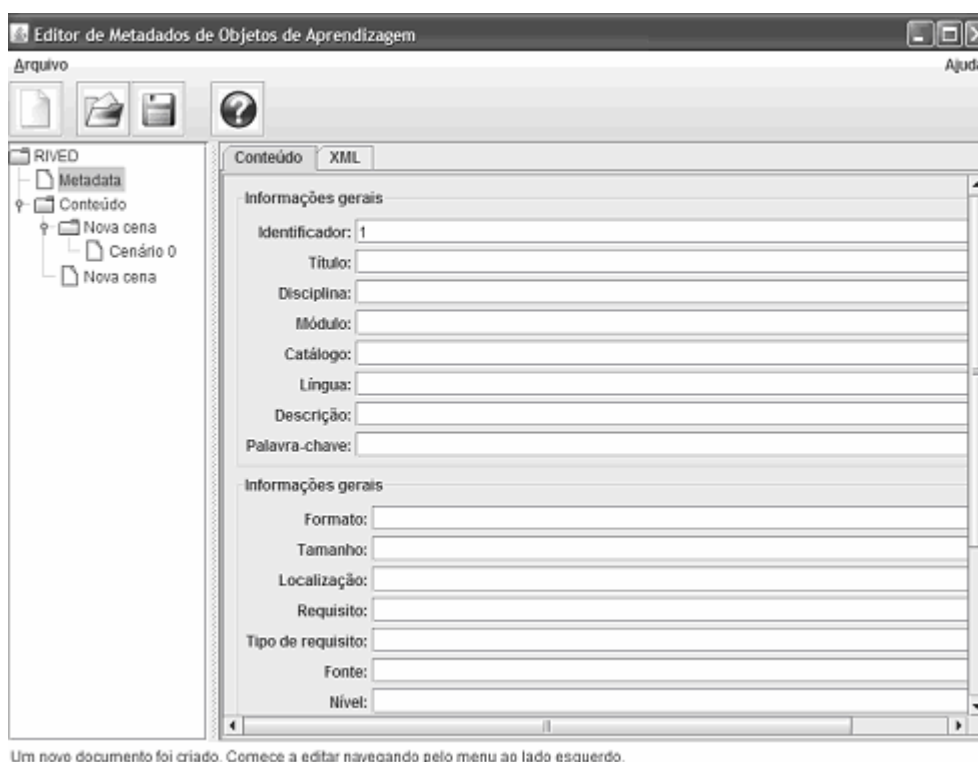


Figura 6.11. Interface que permite editar os metadados LOM

A Figura 6.12 apresenta a interface para edição de conteúdos de um OA. O lado esquerdo apresenta a distribuição do arquivo XML em formato de árvore.

Quando o usuário seleciona um elemento da árvore surge um formulário no lado direito da interface para edição do conteúdo do elemento selecionado. Para gravar na memória, o usuário deve clicar em *Atualiza*. A inserção e/ou remoção de elementos é feita através de um clique com o botão direito do mouse sobre o elemento e selecionando a opção desejada (inserir ou remover).

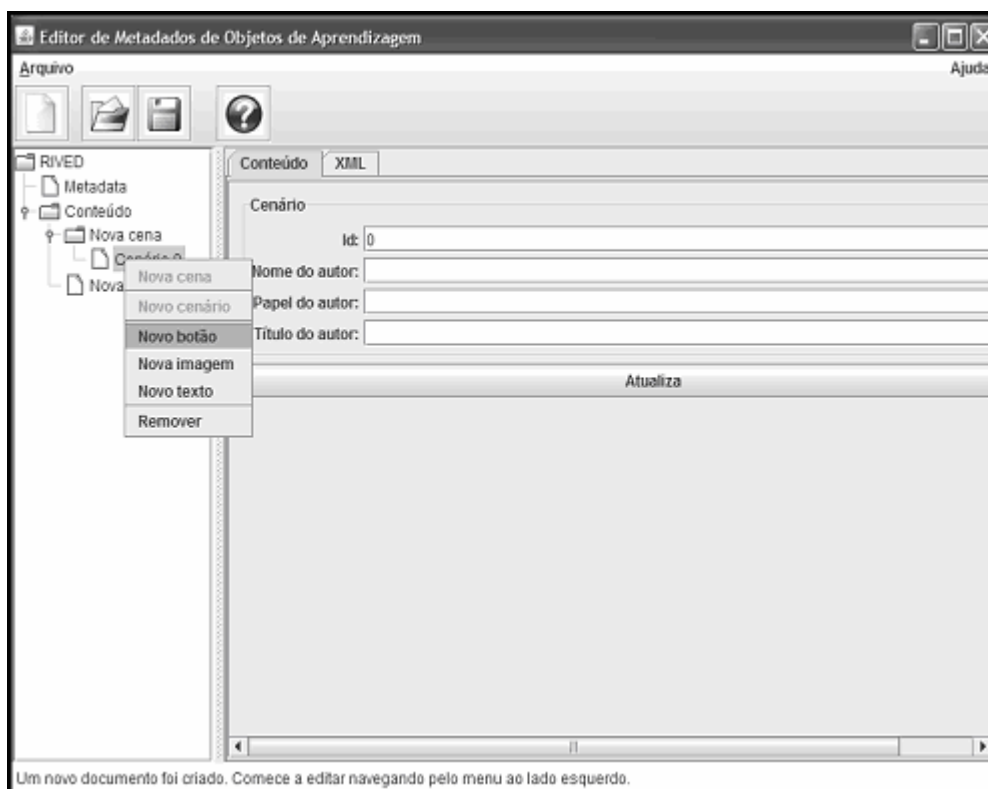


Figura 6.12. Interface da ferramenta desenvolvida

De acordo com os relatos do professor, a ferramenta auxilia bastante a programação do objeto. O professor também apresentou algumas vantagens da ferramenta em relação aos demais editores analisados, conforme mostram os relatos abaixo:

[...] Como ele é um programa que possui a finalidade de desenvolver e modificar objetos, ele disponibiliza diretamente para o programador o padrão típico dos OA: cenários, cenas, textos, imagens, botões. Ele não oferece acesso às *tags*, ou seja, o programador, que pode ser um professor sem muito conhecimento na área de informática, pode alterar um texto sem se preocupar em apagar equivocadamente uma *tag*. Ele disponibiliza para o programador os elementos básicos de um

cenário (texto, imagem e botão), bem como as exigências para que estes elementos sejam inseridos na programação dos OA [Figura 6.13].



Figura 6.13. Interface da ferramenta

O professor, no papel de avaliador, afirma que a ferramenta auxilia na programação de OA e sugere algumas mudanças com o objetivo de torná-la um programa mais completo e que facilite ainda mais a produção de OA. Seguem abaixo as modificações propostas:

- O programa deve disponibilizar a opção voltar para uma ação já realizada;
- O programa deve aceitar que o programador troque cenários/cenas com outros e, além disso, permitir a criação de novos cenários/cenas entre os já existentes (antes ou depois);
- O programa deve permitir a cópia de um botão, imagem ou texto de um cenário para o outro.

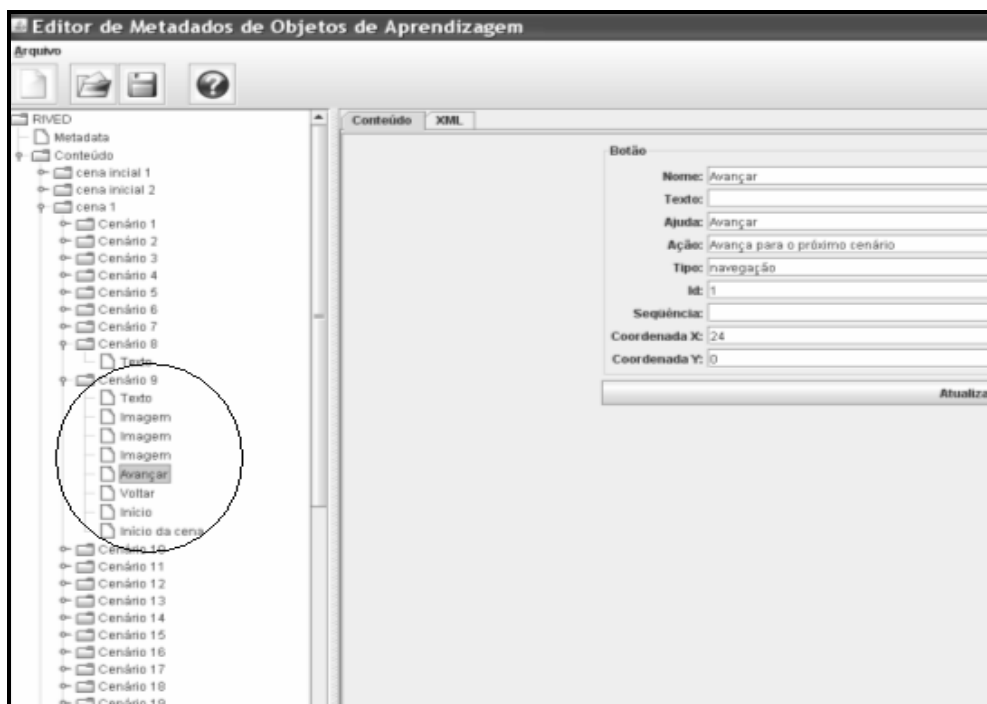


Figura 6.14. Interface da ferramenta- edição do cenário 9

Um exemplo apontado foi utilizando botões. Nesse caso, o botão avançar do cenário 9 (Figura 6.14) é idêntico ao do cenário 45 (Figura 6.15). Dessa forma, poderia ser permitida a cópia do botão avançar do cenário 9 para o cenário 45. Essa funcionalidade de fato está prevista nos casos de uso mas não foi implementada no protótipo da ferramenta.

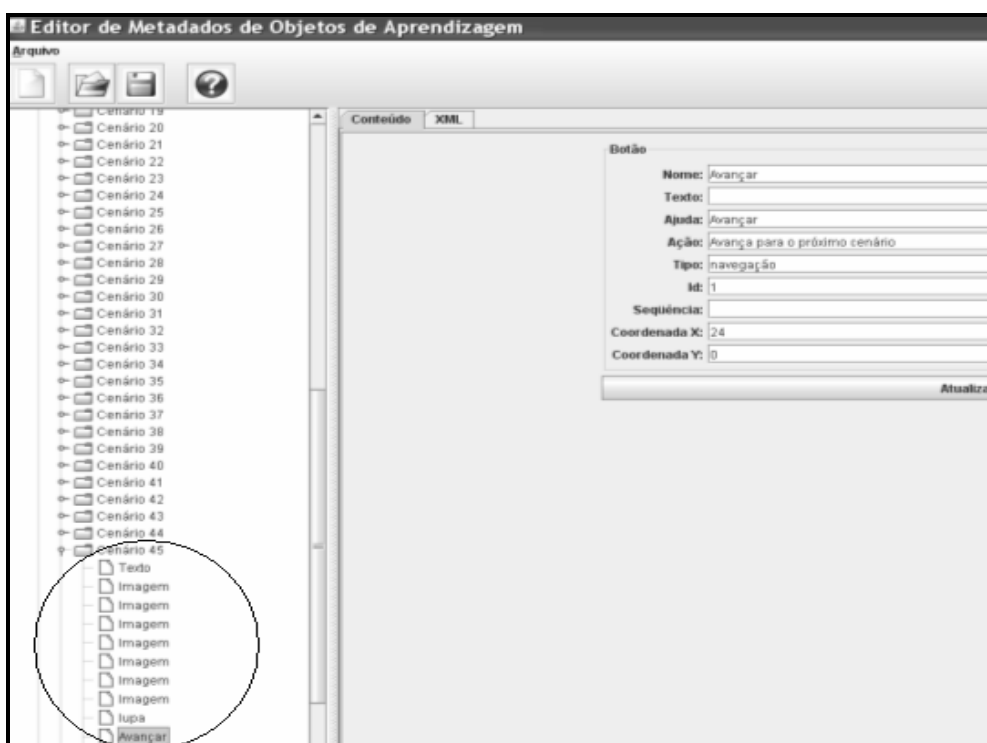


Figura 6.15. Interface da ferramenta – edição do cenário 45

Além de relatar a experiência quanto ao uso da ferramenta, o professor respondeu ao mesmo questionário aplicado para avaliação dos outros editores.

Os dados do questionário foram tabulados usando uma planilha. Conforme descrito na metodologia, o questionário foi dividido em três blocos. Os blocos A e B tiveram peso 3, enquanto o bloco C teve peso 4 por se entender que tais questões eram fundamentais para verificar o diferencial da ferramenta em relação aos editores analisados.

As Figuras 6.16 e 6.17 apresentam o resultado dos dados obtidos dos questionários respondidos.

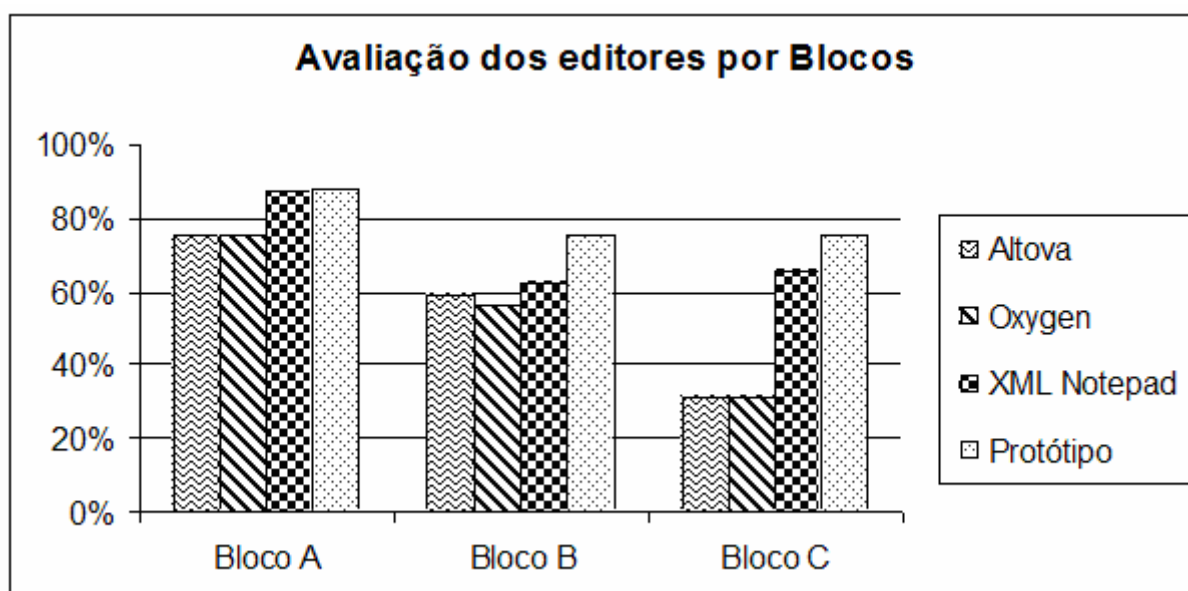


Figura 6.16. Resultado da avaliação por blocos

A Figura 6.16 apresenta o resultado da avaliação dos editores em blocos. Pode-se notar que no bloco A a ferramenta desenvolvida obteve o melhor resultado. Porém, o diferencial em relação ao XML *Notepad* foi pequeno. Já nos blocos B e C, a ferramenta desenvolvida obteve o melhor resultado e a diferença foi maior em relação aos outros editores. Em todos os blocos, a ferramenta obteve o melhor resultado enquanto o editor XML *Notepad* obteve o segundo melhor resultado. O resultado geral (Figura 6.17) contém os resultados dos três blocos.

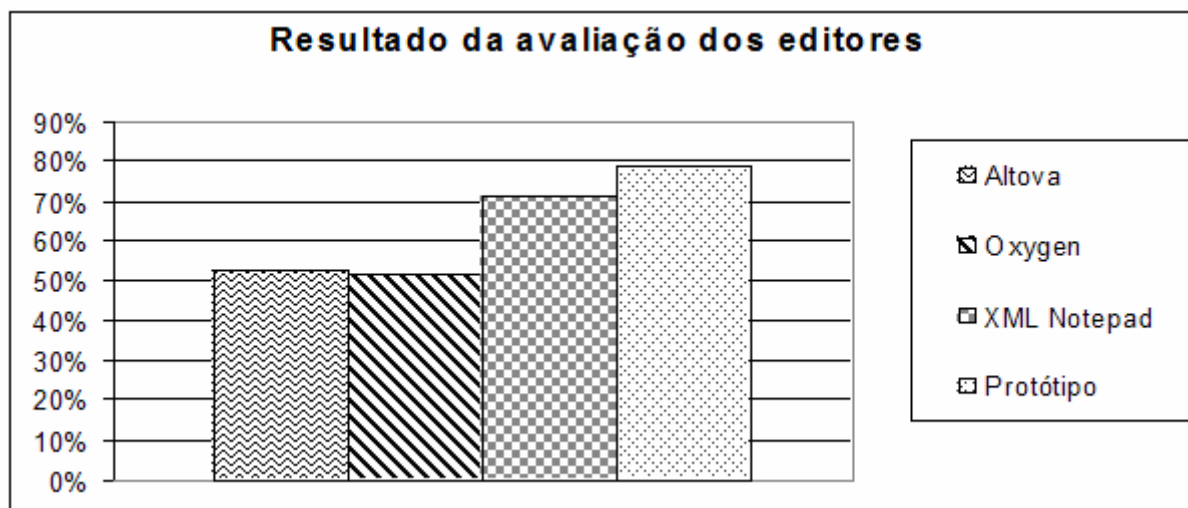


Figura 6.17. Resultado geral da avaliação do editores

De acordo com o resultado geral a ferramenta obteve o melhor resultado. Na segunda posição está o XML *Notepad*, na terceira, o *Altova* e na quarta e última posição, o *Oxygen*.

Os resultados obtidos com a avaliação da ferramenta mostram que os problemas apresentados no Quadro 6.4 foram solucionados conforme o previsto.

Problemas	Resolução dos problemas
O documento XML não possui validação.	A ferramenta permitiu que fosse feita a validação do documento XML. Para quem não entende os detalhes técnicos da linguagem XML essa validação deve acontecer de forma transparente. A ferramenta permite verificar se o arquivo XML está de acordo com o padrão estabelecido.
A hierarquia da estrutura de informação não é padronizada.	O uso da ferramenta permitiu padronizar a hierarquia da estrutura de informação de um OA. É necessário estruturar o documento XML de modo que ele possua partes em comum com outros objetos e ao mesmo tempo permita ter sua particularidade em relação ao conteúdo a ser abordado.
As <i>tags</i> podem ser apagadas equivocadamente.	A ferramenta permitiu que a equipe de domínio altere a parte textual dos OA sem se preocupar com a existência de <i>tags</i> . O modelo de edição é em formato de formulário, onde só o conteúdo pode ser alterado.
As estruturas de informação são diferentes.	A ferramenta permitiu abrir um arquivo XML de modo padronizado. Isso evita que dentro de uma mesma equipe (UENF) existam estruturas diferentes, apesar das diferenças em particular existentes entre cada OA, facilitando a edição por parte da equipe de domínio.

Quadro 6.4. Resolução dos problemas com a criação da ferramenta

As telas correspondentes a cada caso de uso encontram-se no Anexo 7. A ferramenta possui uma aba denominada XML na qual o arquivo XML que está sendo gerado é apresentado, conforme exemplificado na Figura 6.18.

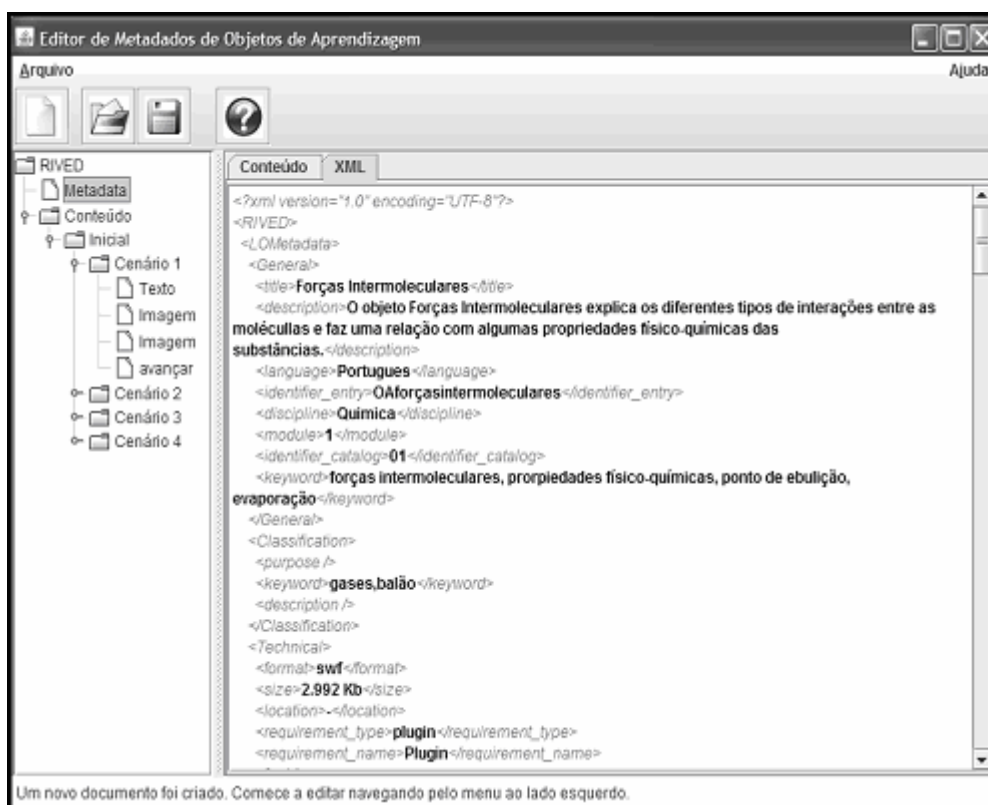


Figura 6.18. Interface da ferramenta- exibição do arquivo XML gerado

A avaliação teve como objetivo verificar também se a ferramenta possui uma interface amigável, de fácil utilização, e que possibilite a edição do conteúdo de forma que as *tags* sejam transparentes ao usuário, assim como a possibilidade de inserção e remoção de elementos, sem que isso cause danos ao conteúdo de um OA. De acordo com as respostas do questionário, chegou-se a conclusão de que a ferramenta possui uma interface de fácil utilização. As questões referentes à facilidade de aprendizado e de operação e controle tiveram resultados entre P+ (Parcialmente com poucas restrições) e S (Sim). Com o uso de formulários para edição, a ferramenta permite que os aspectos técnicos da XML, como o uso de *tags*, sejam imperceptíveis ao usuário, no caso, a uma equipe pedagógica e não técnica.

Sobre a interface da ferramenta o professor relatou:

[...] A interface do *software* desenvolvido no trabalho é amigável e de fácil utilização, principalmente por ser bastante simples. O modelo em árvore utilizado pelo *software* facilita bastante o seu uso, principalmente por dar uma visão geral ao programador.

Ao fim da avaliação observou-se se todos os casos de uso foram implementados e se existiam pendências. O Quadro 6.5 apresenta o resultado dessa análise.

Casos de uso	Implementação e resultado
Abrir arquivo XML	Implementado. Funcionalidade comprovada.
Validar arquivo XML	Implementado. Funcionalidade comprovada. Pendência: Mensagem de erro ao tentar abrir arquivo não válido.
Modificar conteúdo do Arquivo XML	Funciona parcialmente. Pendência: 1- Remover o elemento que estiver vazio (sem conteúdo). 2- Mensagem de erro, caso seja obrigatório o preenchimento do conteúdo do elemento.
Inserir elemento no Arquivo XML	Implementado. Alteração: Não há botão + "nome do elemento".
Remover elemento do Arquivo XML	Funciona parcialmente. Pendência : 1- Se o elemento for obrigatório deve aparecer uma mensagem indicando que não é permitida a remoção. Alteração: Não há botão - "nome do elemento".
Salvar arquivo XML	Implementado. Funcionalidade comprovada.
Salvar como – Arquivo XML	Implementado. Funcionalidade comprovada.

Abrir Novo Editor	Implementado. Funcionalidade comprovada. Alteração: não aparece outro editor. Quando se tem um arquivo aberto e o usuário clica em novo, surge uma mensagem perguntando se quer salvar o arquivo ou não. Se o usuário salvar, surge uma nova interface do editor, caso contrário, não surge uma nova interface.
Fechar Editor	Implementado. Funcionalidade comprovada.
Sair do editor	Implementado. Funcionalidade comprovada.

Quadro 6.5. Casos de uso implementados

Todos os casos de uso foram implementados. Porém, alguns casos de uso implementados demonstram algumas pendências. Os casos de uso Modificar conteúdo, Remover elemento e Abrir novo editor foram alterados na implementação e apresentam algumas pendências.

7. CONCLUSÕES

7.1 Contribuições

Esta dissertação foi desenvolvida no contexto do projeto RIVED/Fábrica Virtual. A experiência da autora no projeto no período de 2004 a 2008, participando da equipe técnica desenvolvendo objetos de aprendizagem na equipe UENF, permitiu conhecer profundamente todo o processo produtivo de OA. A participação de um encontro em Recife em novembro de 2006 permitiu contato com várias outras equipes, de outras universidades brasileiras. A constatação foi que muitos dos problemas existentes no processo utilizado na equipe UENF eram comuns às outras equipes. A partir dessa constatação, foi iniciado o estudo que culminou nesta dissertação visando resolver parte dos problemas existentes. O foco definido para esta definição foi na edição da parte textual dos objetos e dos metadados necessários para cadastramento dos objetos no repositório do RIVED localizado em http://rived.mec.gov.br/site_objeto_lis.php. A persistência dos objetos neste repositório não foi tratada nesta dissertação. Sobre esse tema, consultar Cordeiro (2009).

Na impossibilidade de tratar todas as áreas de domínio abrangidas pelo projeto (Química, Física, Matemática, Geografia, etc.), optou-se por restringir o foco nas equipes de Química, considerando-se que esta é a área da equipe de domínio na qual atuou a equipe RIVED/UENF de 2004 a 2008. A primeira etapa consistiu em analisar a estrutura de informação dos objetos das diferentes equipes, isto é, a estrutura que contém elementos principais que os descrevem como um todo, e outros elementos que descrevem os recursos existentes em cada uma de suas cenas. Como resultado dessa análise, foi constatado que apesar de haver recomendações explícitas por parte do MEC relativas a alguns aspectos (cores das bordas diferenciadas por área de domínio, tamanho do palco⁹, etc.) algumas não eram seguidas. Entre elas, a recomendação para usar uma estrutura externa em XML contendo toda a parte textual dos objetos (notar que algumas equipes incluem os textos junto com o código-fonte). Ainda, foi possível perceber que a forma de apresentação de elementos textuais variava de acordo com a equipe. Finalmente, os elementos textuais utilizados para cadastramento dos objetos no repositório do MEC

⁹ Um palco no *Macromedia Flash* é o espaço no qual são realizadas todas as tarefas de criação e edição. O que estiver dentro do palco é exibido no executável, ao contrário do que estiver fora.

não estavam presentes no próprio objeto de forma explícita, isto é, identificados por um rótulo. Algumas informações encontravam-se presentes de forma implícita, como por exemplo, partes do conteúdo tratado, uma vez que conceitos, explicações e outras aparecem nos objetos. Em outros casos, porém, as informações utilizadas para cadastramento não apareciam no objeto nem implicitamente (por exemplo, informações sobre para qual série do ensino médio o objeto em princípio é destinado).

Além da constatação da falta de padronização dos elementos textuais e também da forma de representação da mesma (interna ou externa ao código-fonte), percebeu-se que o fato da equipe técnica criar e editar os arquivos XML (nos casos das equipes em que as partes textuais são externas ao código-fonte) trazia alguns inconvenientes. Entre eles: i) o aumento do número de ciclos de interação equipe técnica /equipe de domínio para correção do objeto; ii) o retrabalho, uma vez que as partes textuais do objeto são especificadas no roteiro pela equipe de domínio e passadas para o arquivo XML pela equipe técnica. Nesse contexto, a **hipótese** levantada no âmbito dessa dissertação é que passar a criação e edição das partes textuais dos objetos para a equipe de domínio diminuiria o tempo de desenvolvimento de um objeto uma vez que diminuiriam os ciclos de interação entre as equipes e diminuiria o retrabalho. Convergente com esta hipótese, o **objetivo** estabelecido para a dissertação foi otimizar o processo de desenvolvimento de OA através da diminuição do tempo de desenvolvimento passando a criação e edição da parte textual dos objetos para a equipe de domínio.

Para o objetivo ser alcançado foi necessário padronizar a estrutura de informação através da criação de um modelo usando diagrama de classes UML e desenvolver um XML *Schema* que representasse tal modelo. Para passar a manipulação das partes textuais dos objetos para a equipe de domínio foi necessário buscar uma ferramenta de edição de documentos XML suficientemente fácil de usar. Uma vez que os editores genéricos analisados (XML *Notepad*, *Oxygen* e *Altova*) não foram bem aceitos pela equipe de domínio, foi necessário desenvolver um editor específico. A linguagem utilizada para desenvolver o editor foi Java e a IDE foi *Eclipse*. Também foi utilizado o JDOM.

Na estrutura de informação anterior, somente os textos eram descritos. A nova estrutura descreve, além dos metadados do LOM, todos os recursos existentes

no OA como imagens, botões, textos, tabelas, exercícios, enfim, tudo que compõe cenas e cenários.

A nova estrutura de informação ficou mais extensa e complexa que a anterior, devido ao fato de que todos os recursos de cada cena e cenário são descritos separadamente, o que não ocorria na estrutura anterior. Isso permite uma reusabilidade maior do OA, uma vez que permite excluir e inserir elementos a um cenário ou cena, sem que isso afete outros cenários e cenas.

O desenvolvimento de todas as partes textuais de uma cena completa do OA “Forças Intermoleculares”, que explica os diferentes tipos de interações entre as moléculas e faz uma relação com algumas propriedades físico-químicas das substâncias, elaborado por um professor do Instituto Federal Fluminense (antigo CEFET Campos), permitiu constatar que o tempo de desenvolvimento de fato diminuiu, uma vez que os ciclos de interação equipe técnica/equipe de domínio foram reduzidos para três. Primeiro, o arquivo XML é criado pela equipe de domínio usando o protótipo do editor e enviado para a equipe técnica para desenvolver o OA incorporando as partes textuais. Segundo, a equipe técnica envia para a equipe de domínio eventuais necessidades de alteração das partes textuais demandadas por dificuldades de implementação (por exemplo, a quantidade de elementos previstos associados às imagens, animações e outros recursos do OA podem extrapolar o tamanho desejado para o objeto). Finalmente, o arquivo XML é enviado para a equipe técnica para teste final. A partir daí, eventuais modificações nas partes textuais de um OA como, por exemplo, para correções de português, acertos conceituais da área de domínio, mudanças na forma de apresentar o conteúdo que não impliquem em mudança de programação passaram a ser feitas diretamente pela equipe de domínio. Essa mudança no processo¹⁰ de desenvolvimento de OA, apresentado abaixo na Figura 7.1, pode ser destacada como a principal contribuição dessa dissertação.

Com a criação e a avaliação da ferramenta pôde-se verificar mudanças no processo produtivo como exposto na Figura 7.1.

¹⁰ A análise do processo produtivo com indicação da possível solução apresentada nesta dissertação foi publicada em artigo no Encontro Nacional de Engenharia de Produção de 2008 (SANTOS, RAPKIEWICZ e WIVES, 2008a).

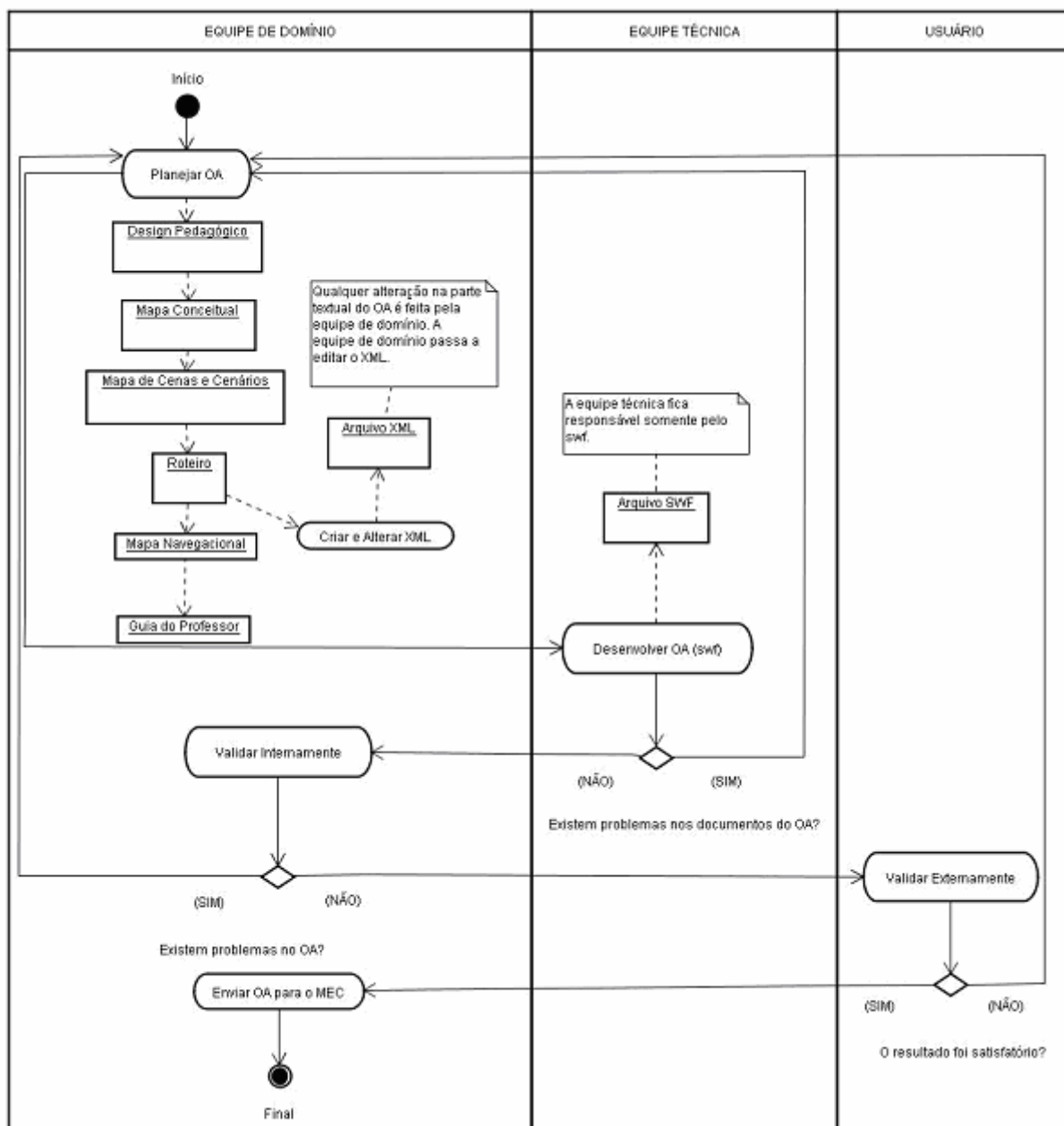


Figura 7.1. Processo produtivo atual-modificado com a criação da ferramenta

No processo produtivo atual a equipe de domínio passou a criar e editar a estrutura de informação, conforme mostra o diagrama de atividades da Figura 7.1, utilizando a ferramenta desenvolvida. A equipe técnica desenvolve somente o swf com base em orientações recebidas da equipe de domínio. Na etapa de planejamento a equipe de domínio desenvolve os documentos: *Design* pedagógico, Mapa conceitual, Mapa de cenas e cenários, Roteiro, Mapa navegacional e o arquivo XML que representa a estrutura de informação. Ao invés de fazer o Roteiro a própria equipe de domínio edita a estrutura de informação usando o editor XML

desenvolvido nesta dissertação. Cabe ressaltar que a interação é constante entre a equipe de domínio e a equipe técnica de modo a evitar erros futuros.

Além da mudança estabelecida no processo produtivo, as várias etapas da dissertação permitiram outras contribuições, entre as quais as principais são:

1. Uma estrutura de informação padronizada para objetos de aprendizagem representada através de um diagrama de classes UML, validada através de uma prova de conceito junto a equipe RIVED/UENF através da re-estruturação¹¹ do arquivo XML do OA “Balão”.

2. Um esquema XML para permitir a validação da estrutura de informação de objetos de aprendizagem. Apesar de ter sido validada na área de Química, é aplicável a outras áreas de domínio.

3. A análise da usabilidade e funcionalidade de editores genéricos¹² para XML *Notepad*, *Oxygen* e *Altova*.

4. O protótipo de um editor XML específico para o modelo proposto padronizando a estrutura de informação.

7.2 Dificuldades Encontradas

Com a padronização da estrutura de informação para OA em XML, ao tentar adaptar os objetos a essa nova estrutura algumas dificuldades foram encontradas. A nova estrutura de informação descreve além dos recursos existentes no OA, os metadados relativos a descrição do mesmo baseado no padrão LOM. Esses metadados estão divididos em 9 categorias. Uma das dificuldades encontradas foi justamente descrever algumas informações acerca dos OA que no modelo anterior não foram descritas. Ou seja, recuperar informações que não estão explícitas em nenhum dos documentos produzidos durante o desenvolvimento dos objetos.

Outra dificuldade encontrada foi encontrar uma descrição para os elementos que estão sempre presentes no OA, independente da cena e do cenário. As diferentes propostas de nomes sempre deixavam a desejar devido a diversidade de elementos de caráter geral que podem estar presentes nos objetos (mapas conceituais, exercícios, créditos). Por fim, optou-se por denominar o grupo desses elementos de *CommonsElementsScene*, ou seja, elementos comuns às cenas.

¹¹ A proposta deste modelo foi publicada em artigo no *Workshop* de Informática na Escola (WIE) da Sociedade Brasileira de Computação, Belém, 2008 (SANTOS *et al.*, 2008).

¹² O resultado dessa análise foi publicado como *short paper* no Simpósio Brasileiro de Informática na Educação em Fortaleza, 2008 (SANTOS, RAPKIEWICZ e WIVES, 2008b).

Outras dificuldades encontradas na descrição de recursos dos OA foram em relação a categorização como, por exemplo, existem vários tipos de botões e de texto, e no modelo anterior não havia necessidade desse tipo de especificação.

Finalmente, dificuldades de ordem técnica foram encontradas. A ausência de corpo docente na UENF que atua nas áreas abrangidas pela dissertação e outros projetos correlatos limitou a troca de experiências durante o mestrado que normalmente ocorre entre vários colegas de curso. Por outro lado, essa dificuldade propiciou integração com o professor do Instituto de Informática da UFRGS. Após um contato presencial de uma semana em novembro de 2007, em Porto Alegre, todo contato com o mesmo e reuniões entre a equipe foi feita através do MSN e *Skype*. A vivência do uso de recursos de TIC no processo educativo em nível de mestrado mostrou que é possível ampliar o escopo de atuação na pesquisa quando se usam os recursos tecnológicos existentes para comunicação e colaboração.

Finalmente, uma dificuldade constatada foi a eliminação do roteiro no processo de desenvolvimento. Em princípio, parte significativa do mesmo foi substituída pela edição da estrutura de informação. No entanto, a explicação de como esses textos devem ser trabalhados pela equipe técnica durante a programação do objeto permanecem em aberto. Cumpre ressaltar, no entanto, que esta dificuldade já era observada no próprio roteiro, uma vez que a equipe de domínio tende a não explicitar e detalhar essas explicações.

7.3 Trabalhos Futuros

Como continuação do modelo criado nessa dissertação pode ser desenvolvida uma ontologia para a estrutura de informação. Segundo Borst (1997, p.12) a definição de ontologia é “*uma especificação formal e explícita de uma conceitualização compartilhada*”. A importância do uso de ontologias é a capacidade de melhorar os processos de recuperação de informação ao organizar o conteúdo de fontes de dados em um determinado domínio (ALMEIDA e BAX, 2003).

Outro trabalho que pode ser feito é a adaptação da interface da ferramenta para a interface Padrão. A interface Padrão é uma interface modelo para OA. Essa interface abrange todos os recursos da estrutura de informação padrão e sugere a disponibilização desses recursos de forma a atender a critérios de usabilidade. A Figura 7.2 apresenta o OA “Balão” no modelo da Interface Padrão.



Figura 7.2. Exemplo da Interface Padrão

Dessa forma o usuário criaria a estrutura de informação através dessa interface, inserindo e removendo os recursos que deseja e os textos já apareceriam em seus locais apropriados. Poderia acrescentar cenas e cenários e ainda os textos, imagens, botões, tabelas e exercícios e ao final, a estrutura de informação estaria criada de acordo com os recursos selecionados na montagem da interface.

Outra pesquisa seria o desenvolvimento de uma ferramenta que permita a criação não somente da estrutura de informação dos OA, mas também a montagem de OA. Ou seja, o usuário poderia criar o objeto, com as imagens, botões, animações e textos que desejasse. Cada parte de um OA, seria, por si própria, um objeto com granularidade pequena. Esse objeto poderia ter formas e modelos diferentes e o usuário ainda poderia criar o seu objeto ou importá-lo de outro local.

O fato da estrutura de informação estar padronizada facilita o processo no que concerne a equipe de domínio. Por outro lado, no que concerne a equipe técnica, o potencial de estruturação da XML é perdido mantendo-se a forma de tratar a parte textual, na programação do objeto, através do armazenamento em vetores. É necessário, pois, desenvolver uma forma de manipulação direta da árvore XML no processo de desenvolvimento de objetos no *Action Script* e em outras linguagens usadas pelas equipes no desenvolvimento de OA. O problema relacionado com a apresentação de textos que necessitam de subscripto ou sobrescrito, como fórmulas químicas, não foi abordado nessa dissertação, considerando-se que não afeta o modelo (estrutura de informação).

Finalmente, o aperfeiçoamento do editor, uma vez que o que foi desenvolvido na dissertação limitou-se a um protótipo.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC: 9126-1 *Engenharia de software - Qualidade de produto Parte 1: Modelo de qualidade*, Jun. 2003.

_____. *NBR ISO 9000: Sistemas de gestão da qualidade - Fundamentos e Vocabulário*, Rio de Janeiro - RJ, 26 p. 2000.

ABREU, Marlon Freitas; CORDEIRO, Rogério Avellar; RAPKIEWICZ, Clevis Elena; CANELA, Maria Cristina, Utilizando Objetos de Aprendizagem no Processo de Ensino e Aprendizagem de Química no Ensino Médio: o Caso dos Óxidos e da Poluição Atmosférica, *WIE XII - Workshop de Informática na Escola*, Campo Grande-MS, p. 336-344. 2006.

ADL, *Advanced Distributed Learning – SCORM*, 2008. Disponível em: <<http://www.adlnet.gov/scorm/index.aspx>>. Acesso em: 10 Out. 2008.

ALMEIDA, Mauricio Barcellos, Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares, *Ciência da Informação*, Brasília, v. 31, n. 2, p. 5-13, maio/ago. 2002.

ALMEIDA, Maurício Barcellos; BAX, Marcello Peixoto, Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção, *Ciência da Informação*, Brasília, v. 32, n. 3, p. 7-20, set./dez. 2003

ALVES, Maria das Dores Rosa; SOUZA, Marcia Izabel Fugisawa, Estudo de correspondência de elementos metadados :Dublin Core e Marc 21, *Revista Digital de Biblioteconomia e Ciência da Informação*, Campinas, v. 4, n. 2, p. 20-38, jan./jun. 2007.

AMARAL, Lucas Lopes; GOMES, Thiago; SOUZA, Maria de Fátima; CASTRO FILHO, José Aires; PEQUENO, Mauro, Um aprimoramento do modelo de processo de criação de objetos de aprendizagem do projeto RIVED, *WIE - XII Workshop de Informática na Escola*, Campo Grande - MS, p. 373-376. 2006.

BASSANI, Patricia Scherer; PASSERINO, Liliana; PASQUALOTTI, Paulo; RITZEL, Marcelo Iserhardt, Em busca de uma proposta metodológica para o desenvolvimento de *software* educativo colaborativo, *RENOTE- Revista Novas Tecnologias na Educação*, Porto Alegre, V. 4 1:1-10, Julho. 2006.

BAX, Marcello Peixoto, Introdução às linguagens de marcas, *Ciência da Informação*, Brasília, v. 30, n. 1, p. 32-38, jan./abr. 2001.

BORST, Willem Nico, *Construction of engineering ontologies for knowledge sharing and reuse*. 1997. Tese (Doutorado). Disponível em: <<http://doc.utwente.nl/17864/1/t0000004.pdf>>. Acesso em: 03 Dez. 2008.

BRAGANHOLA, Vanessa; HEUSER, Carlos, XML Schema, RDF(S) e UML: uma comparação, *IDEAS - 4th Iberoamerican Workshop on Requirements Engineering and Software Environments*, Santo Domingo, Heredia, Costa Rica. p. 78-90. 2001.

BRITO, Ronnie Fagundes; PEREIRA, Alice Theresinha Cybis; BRAGA, Marta Goulart, Desenvolvendo Objetos de Aprendizagem SCORM aplicando a Arquitetura da Informação e Teoria da Flexibilidade Cognitiva, *Informática na Educação: teoria & prática*, Porto Alegre, v.9, n.1, jan./jun. 2006.

CANELA, Maria Cristina; RAPKIEWICZ, Cleli Elena; SANTOS, Angélica Freitas; SILVA, Fabio Pereira; ABREU, Marlon Freitas; SANTOS, Núbia dos Santos Rosa Santana; CORDEIRO, Rogério Avellar; COSTA, Valeria Machado, A educação ambiental no ensino de química através de objetos de aprendizagem, In: Souza Jr, A. J., Fernandes, M.A.; Lopes, C. R.; Ghisolfi da Silva, R. M.; (Org.). *Informática na Educação: elaboração de objetos de aprendizagem*. Uberlândia: EDUFU - Editora da Universidade Federal de Uberlândia, p. 13-37, 2007.

CASTELLS, Manuel , *A Era da Informação: economia, sociedade e cultura – Fim de Milênio* , São Paulo: Paz e Terra, v. 3 , p.426, 1999.

CORDEIRO, Rogério Avellar Campos, *FPOA: Um framework de persistência de objetos de aprendizagem para catalogação em repositórios*, Dissertação (Mestrado em Engenharia de Produção), Rio de Janeiro, Universidade Estadual do Norte Fluminense Darcy Ribeiro - UENF, 2009.

CORDEIRO, Rogério Avellar Campos; RAPKIEWICZ, Cleli Elena; CANELA, Maria Cristina; SANTOS, Angélica Freitas; CARNEIRO, Eduardo Coelho, Utilizando mapas conceitual, de cenário e navegacional no apoio ao processo de desenvolvimento de objetos de aprendizagem, *RENOTE - Novas Tecnologias na Educação*, Porto Alegre, v.5 1:1-14. Julho. 2007.

COSTA, Valeria Machado; RAPKIEWICZ, Cleli Elena; CANELA, Maria Cristina, Avaliando os objetos de aprendizagem da Fábrica Virtual: uma proposta de metodologia, *Encontro Internacional de Inovações Pedagógicas*, Recife, 2006.

DEITEL, Harvey; DEITEL, Paul, *Java - como programar*, 4. ed. Porto Alegre: Bookman, 2003.

DUTRA, Renato; TAROUCO, Liane; PASSERINO, Liliana, O Uso de Objetos de Aprendizagem SCORM para apoiar a Avaliação Formativa, *XIX SBIE - Simpósio Brasileiro de Informática na Educação*, Fortaleza-CE. 2008.

FONSECA, Rúben; SIMÕES Alberto, Alternativas ao XML: YAML e JSON, XATA2007 : XML : aplicações e tecnologias associadas, *V Conferência Nacional em XML*, p. 33–46, Lisboa, 2007.

FOWLER, Martin , *UML Essencial: Um breve guia para a linguagem padrão de modelagem de objetos*, Porto Alegre: Bookman, 2005.

HAROLD, Elliotte Rusty, *Processing XML with Java: a guide to SAX, DOM, JDOM, JAXP and TrAX*, Addison-Wesley , 2003.

HODGE, Gail. *Metadata made simpler: a guide for libraries*, Bethesda, MD: National Information Standards Organizations, 15p, 2001.

IEEE/LTSC - Learning Technology Standards Committee, *Standard for Learning Object Metadata*, 2003, Disponível em: <<http://ltsc.ieee.org/wg12/>>. Acesso em: 10 Jun. 2008.

JSON, *Introdução ao JSON*, 2008. Disponível em: <<http://www.json.org/json-pt.html>> . Acesso em: 18 nov. 2008.

LARMAN, Craig, *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado*, 2.ed. Porto Alegre: Bookman, 607 p., 2004.

- LIMA, Gercina Ângela Borém. Mapa Conceitual como ferramenta para organização do conhecimento em sistema e hipertextos e seus aspectos cognitivos, *Perspectivas em Ciência da Informação*, Belo Horizonte, v.9 n.2, p. 134-145, jul./dez. 2004.
- LONGMIRE, Warren, *A Primer On Learning Objects*, American Society for Training & Development. Virginia. USA, 2000. Disponível em:<http://www.astd.org/LC/2000/0300_longmire.htm>. Acesso em: 11 Dez. 2008.
- MALHOTRA, Naresh, *Pesquisa de marketing: uma orientação aplicada*, 4. ed., Porto Alegre: Bookman, 2006.
- MALLARD, Romain, *Interoperabilidade dos Conteúdos Didáticos Digitais: uma contribuição a questão dos padrões*, Dissertação (Mestrado em Informática Aplicada), Curitiba-PR, Pontifícia Universidade Católica do Paraná- PUCPR, 145 p. 2004.
- MARCHAL, Benoit, *XML by example*. QUE, 2000.
- MATTAR, Fauze Najib, *Pesquisa de Marketing*. São Paulo: Editora Atlas, 1997.
- MELLO, Ronaldo Santos; DORNELES, Carina Friedrich; KADE, Adrovane; BRAGANHOLO, Vanessa de Paula; HEUSER, Carlos Alberto, Dados Semi-Estruturados, *XV Brazilian Database Symposium*, João Pessoa- PB, 39 p. 2000.
- NASCIMENTO, Anna; MORGADO, Eduardo, *Um projeto de colaboração internacional na América*, 2003. Disponível em <<http://www.rived.mec.gov.br/artigos/rived.pdf> >. Acesso em: 28 Nov. 2008.
- NASCIMENTO, Helio; SILVA, Charlyton, XML e o seu poder de interoperabilidade, *Revista Eletrônica da FAL-Faculdade de Alagoas*, n.1, p.34-45, 2007.
- NOVAK, Joseph David. *The theory underling concept maps and how to construct them*. 1984. 11p. Disponível em <<http://cmap.coginst.uwf.edu/info/> >. Acesso em: 03 Dez. 2008.
- NOVAK, Joseph David; CANÂS , Alberto, The Theory Underlying Concept Maps and How to Construct Them, *Technical Report IHMC CmapTools*, Florida Institute for Human and Machine Cognition, 2006.
- OGDL, *The OGD L Specification*, 2005. Disponível em: <<http://ogdl.sourceforge.net/spec/>>. Acesso em: 20 Jul. 2008.
- PACHECO, Roberto Carlos Santos; KERN, Vinícius Medina, Uma ontologia comum para a integração de bases de informações e conhecimento sobre ciência e tecnologia, *Ciência da Informação*, Brasília, v. 30, n. 3, p. 56-63, set./dez. 2001.
- PRATA, Carmem Lucia, Estratégias para desenvolvimento de conteúdo, In: *III Congresso de Tecnologia Educacional - TecEducation*, 2006. Disponível em:<<http://www.techne.com.br/humus/download/BLOCO%20TEMATICO%202/D%20-%20Carmem%20Prata.pdf>>. Acesso em: 20 Out. 2008.
- PRESSMAN, Roger, *Engenharia de Software*, 6.ed., São Paulo: McGraw-Hill, 2006.
- PRETTO, Nelson de Luca; SERPA, Luis Felipe Perret, A educação e a sociedade da informação, *II Conferencia Internacional Challenges*, 2001.
- RIVED, *Rede Interativa Virtual de Educação a Distância*, 2007. Disponível em:<<http://www.RIVED.mec.gov.br/>>. Acesso em: 3 Nov. 2007.
- _____. *Rede Interativa Virtual de Educação*, 2008. Disponível em:<<http://www.rived.mec.gov.br/aceso.php>> . Acesso em: 10 Set. 2008.

SANTANCHÊ, André; LAGO, Andréa, Objetos digitais complexos na educação e os objetos de aprendizagem, In: *XVIII Simpósio Brasileiro de Informática na Educação-SBIE*, Anais do XVIII Simpósio Brasileiro de Informática na Educação, Mini Cursos, São Paulo, 2007

SANTANCHÊ, André; TEIXEIRA, Cesar Augusto Camillo, Explorando Linguagens de Markup Extensíveis na Construção de Sistemas de Educação Baseados na Web, *XXVI SEMISH - Seminário Integrado de Software e Hardware*, Rio de Janeiro-RJ, v. 1. p. 39-55. 1999.

SANTOS, Fátima. Movimento e interatividade na web. *Publish*, v. 8, n. 55, p. 24-25, jul./ago. 2001.

SANTOS, Núbia dos Santos Rosa Santana; RAPKIEWICZ, Clevis Elena; WIVES, Leandro Krug., O processo produtivo de objetos de aprendizagem numa unidade do RIVED/ Fábrica Virtual: problemas e soluções. In: *XXVIII Encontro Nacional de Engenharia de Produção (ENEGEP)*, Rio de Janeiro, 2008a.

_____. Uma análise da estrutura de informação de Objetos de Aprendizagem. In: *Simpósio Brasileiro de Informática na Educação - SBIE*, Anais do XIX Simpósio Brasileiro de Informática na Educação, Fortaleza, 2008b.

SANTOS, Núbia dos Santos Rosa Santana; CORDEIRO, Rogério Avellar; RAPKIEWICZ, Clevis Elena; WIVES, Leandro Krug, Uma proposta de modelo para Objetos de Aprendizagem. In: *Workshop sobre Informática na Escola - WIE*, XXVIII Congresso da Sociedade Brasileira de Computação, Belém, 2008.

SCHEER, Sergio, GAMA, Carmem Lúcia Graboski, Construção de um repositório para objetos educacionais hiperídia, In: *Congresso Nacional de Ambientes Hiperídia para Aprendizagem – CONAHPA*, Florianópolis - SC, 8 p. 2004.

SEBESTA, Robert, *Conceitos de linguagens de programação*. 4. ed. Porto Alegre: Bookman, 2000.

SEEDNET, *Revista Eletrônica de Educação a Distância*, 2007. Disponível em: <<http://www.seednet.mec.gov.br/noticias.php?codmateria=3218&frmcodprograma=>>. Acesso em: 20 Out. 2007.

SETZER, Valdemar, Dado, informação, conhecimento e competência. *DatagramaZero - Revista de Ciência da Informação*, Rio de Janeiro, n.0, dez. 1999. Disponível em: <http://www.dgz.org.br/dez99/Art_01.htm>. Acesso em: 06 Nov. 2008.

SGML, *Gentle Introduction to SGML*, 2005. Disponível em: <<http://www.w3.org/MarkUp/SGML/>>. Acesso em: 28 Nov. 2008.

SILVA, Janete Fernandes; FERREIRA, Marta Araújo Tavares; BORGES, Mônica Erichsen Nassif. Análise metodológica dos estudos de necessidades de informação sobre setores industriais brasileiros: proposições. *Ciência da Informação*, Brasília, v. 31, n. 2, p. 129-141, 2002.

SILVA, Rejane Maria Ghisolfi; FERNANDES, Márcia Aparecida, Produção e Desenvolvimento de Objetos de Aprendizagem para o ensino de Química: implicações na formação docente, In: Souza Jr, A. J., Fernandes, M.A.; Lopes, C. R.; Ghisolfi da Silva, R. M.; (Org.). *Informática na Educação: elaboração de objetos de aprendizagem*. Uberlândia: EDUFU - Editora da Universidade Federal de Uberlândia, 2007, p. 61-74.

SOUZA, Marcia Izabel Fugisawa; VENDRUSCULO, Laurimar Gonçalves; MELO, Geane Cristina, Metadados para a descrição de recursos de informação eletrônica: utilização do padrão Dublin Core. *Ciência da Informação*, v. 29, n. 1, p. 93-102, abr. 2000.

SOUZA, Maria de Fátima Costa; ANDRADE, Rossana; CASTRO FILHO, José Aires, Aplicando Engenharia de Software Orientado a Modelos ao Desenvolvimento de Objetos de Aprendizagem. In: *XIII Workshop de Teses e Dissertações em Engenharia de Software - XXII Simpósio Brasileiro de Engenharia de Software*, Campinas, p. 62-69, 2008.

SOUZA, Renato Rocha; ALVARENGA, Lídia, A Web Semântica e suas contribuições para a ciência da informação. *Ciência da Informação*, Brasília, v. 33, n. 1, 2004.

TAKAHASHI, Tadao (Org) *Sociedade da Informação no Brasil: Livro Verde*, Ministério da Ciência e Tecnologia, Brasília, 195 p., 2000.

TAROUCO, Liane Margarida Rockenbach; FABRE, Marie-Christine Julie Mascarenhas; TAMUSIUNAS, Fabrício Raupp, Reusabilidade de objetos educacionais. *IX Ciclo de Palestras Novas Tecnologias na Educação. RENOTE* v.1 n.1. 2003.

TAVARES, Romero, Aprendizagem significativa, codificação dual e objetos de aprendizagem, In: *Congresso de Ensino Superior a Distância- IV ESUD – Brasília*, 2006.

TOLENTINO, Ricardo José Vaz; MOITA, Gray Farias, Aplicações WEB em XML: estágio atual e tendências futuras, *Educação e Tecnologia*, Belo Horizonte, v.7, n.1, p.61-71, jan./jun. 2002

VAHLDICK, Adilson ; RAABE Andre, Adaptação de Conteúdo SCORM em Ambientes Inteligentes de Aprendizagem Adaptação de Conteúdo SCORM em Ambientes Inteligentes de Aprendizagem, In: *Simpósio Brasileiro de Informática na Educação- SBIE*, Anais do XIX Simpósio Brasileiro de Informática na Educação, Fortaleza, 2008.

VELOSO, Renê Rodrigues, *Java e XML - Processamento de documentos XML com Java*, 2.ed. São Paulo: Novatec, 109 p. 2007.

VIEIRA, Humberto; RUBERG, Gabriela; MATTOSO, Marta, XVerter: Armazenamento e Consulta de Dados XML em SGBDs, In: *XVII Simpósio Brasileiro de Banco de Dados*, 2002.

W3C, *Extensible Markup Language*, 2007. Disponível em:<<http://www.w3.org/XML/>>. Acesso em: 16 Jul. 2008.

_____. *Document Object Model (DOM) Level 2 Specification*, 2000. Disponível em:<<http://www.w3.org/TR/2000/CR-DOM-Level-2-20000510/DOM2.pdf>>. Acesso em: 10 Jan. 2009.

W3SCHOOLS, *XSLT Tutorial*, 2008. Disponível em:<<http://www.w3schools.com/xsl/default.asp>>. Acesso em: 21 Jan. 2009.

_____. *W3Schools Online Web Tutorials*, 2007. Disponível em:<http://www.w3schools.com/Schema/Schema_intro.asp>. Acesso em: 28 Ago. 2008.

WARPECHOWSKI, Mariúsa, *Recuperação de Metadados de Objetos de Aprendizagem no AdaptWeb*, Dissertação (Mestrado em Ciências da Computação), Porto Alegre-RS, Universidade Federal do Rio Grande do Sul - UFRGS, 69 p. 2005.

WERNER, Claudia; BRAGA, Regina; ZOPELARI, Mônica; BARROS, Márcio; MURTA, Leonardo, *Odyssey-LE: Uma Infra-Estrutura de Reutilização para o Domínio de Processamento Legislativo*, V *ENIAL – Encontro Nacional de Informática Aplicada ao Legislativo*, Vitória –ES, 2000

WILEY, David, *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*, 2002. Disponível em: <<http://reusability.org/read/chpters/wiley.doc>>. Acesso em: 10 Nov. 2007.

_____. *Learning Object Design and sequencing theory*, 2000. Disponível em <<http://www.reusability.org>> . Acesso em: 10 Dez. 2007.

_____. *Open Educational Resources: The Future of Learning Objects*, Palestra Internacional, In: *XIX Simpósio Brasileiro de Informática na Educação-SBIE*, Fortaleza, 2008.

WYSOCKI, Robert, *Effective Project Management: Traditional, Adaptive, Extreme*. 4a edição. John Wiley & Sons, 2006.

ANEXO 1: Formulário de cadastramento do OA Balão



Formulário de cadastramento		
Não altere a formatação dos campos.		
AUTORIA		
Instituição: Universidade Estadual do Norte Fluminense Darcy Ribeiro		Ano: 2007
Autor(es): Coordenação Tecnológica: Clevi Elena Rapkiewicz, Coordenação Pedagógica: Maria Cristina Canela, D.Sc. Design e Modelagem: Glaucia Ribeiro Gonzaga e Marcelo Eichler, D.Sc. Desenvolvimento: Eduardo Coelho Carneiro Supervisão visual e avaliação: Valéria Machado Costa		e-mail: clevi@uenf.br; mccanela@uenf.br
Título do conteúdo/objeto: Conhecendo os gases em uma corrida de balão		
Área(s) de Conhecimento: <input type="checkbox"/> Geografia <input type="checkbox"/> Matemática <input type="checkbox"/> Física <input type="checkbox"/> Biologia <input checked="" type="checkbox"/> Química <input type="checkbox"/> História <input type="checkbox"/> Línguas <input type="checkbox"/> Outro:	Nível (is): <input type="checkbox"/> Ensino Fundamental <input checked="" type="checkbox"/> Ensino Médio <input type="checkbox"/> Ensino profissionalizante <input type="checkbox"/> Ensino Superior <input type="checkbox"/> Atendimento às necessidades especiais	
Série: Ensino Fundamental <input type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> 3° <input type="checkbox"/> 4° <input type="checkbox"/> 5° <input type="checkbox"/> 6° <input type="checkbox"/> 7° <input type="checkbox"/> 8° Médio <input checked="" type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> 3° Ensino profissionalizante <input type="checkbox"/> Ensino Superior <input type="checkbox"/> Atendimento às necessidades especiais <input type="checkbox"/>	Tecnologia(s) utilizada(s): <input type="checkbox"/> HTM <input checked="" type="checkbox"/> Flash <input type="checkbox"/> Java <input type="checkbox"/> PHP <input type="checkbox"/> ASP <input checked="" type="checkbox"/> XML <input type="checkbox"/> Outro:	
Conteúdo/objeto: <input type="checkbox"/> Módulo RIVED <input checked="" type="checkbox"/> Atividade RIVED <input type="checkbox"/> Conteúdo produzido para o PAPED <input type="checkbox"/> Conteúdo produzido para o Prêmio RIVED <input type="checkbox"/> Outro:	Documentação do conteúdo: <input type="checkbox"/> Arquivos completos do conteúdo <input checked="" type="checkbox"/> Guia do professor - PDF ou zipado	Licenciado: <input checked="" type="checkbox"/> Creative Commons <input type="checkbox"/> Outro. Informar qual..... ...
Identificação do conteúdo/objeto: <input type="checkbox"/> Simulação <input type="checkbox"/> Animação <input checked="" type="checkbox"/> Animação e simulação <input type="checkbox"/> Site e animação <input type="checkbox"/> Site e simulação <input type="checkbox"/> Site e animação e simulação <input type="checkbox"/> Outro:	Caso o conteúdo/objeto de aprendizagem seja originado de outro conteúdo informar:.....	
Tipo de conteúdo: (motivação, observação, reconhecimento, análise, experimentação, resolução de problemas, previsão, prática, exploração, discussão etc). Observação, previsão e motivação		



<p>Objetivos de aprendizagem: <i>(habilidades e competências).</i></p> <ul style="list-style-type: none"> ▪ Identificar formas de alterar a altitude de um balão; ▪ Explicar por que um balão de balonismo flutua no ar; ▪ Analisar a relação existente entre aumento de altitude e diminuição da pressão; ▪ Descrever a propriedade responsável por fazer o balão subir; ▪ Comparar as densidades dos gases dentro e fora do balão, estabelecendo suas diferenças; ▪ Calcular a densidade dos gases de diferentes maneiras; ▪ Compreender os princípios básicos da termodinâmica que regem os gases; ▪ Distinguir os comportamentos dos gases de acordo com os valores de suas densidades e a variação de temperatura. ▪ Observar as diferentes características dos diferentes gases e no que estas influenciam seu comportamento. ▪ Reconhecer as principais aplicações dos gases nos lares, indústrias, hospitais, etc. ▪ Compreender e relacionar os principais fatores que influenciam no comportamento dos gases (P, T, V e n)
<p>Conceitos: <i>(conceitos trabalhados na atividade).</i></p> <p>Comportamento dos gases, cinética dos gases, densidade, pressão e temperatura.</p>
<p>Pré-requisitos: <i>(conhecimentos prévios do aluno necessários para a atividade).</i></p> <ul style="list-style-type: none"> ▪ Hipótese de Avogadro ▪ Cálculos matemáticos básicos
<p>Característica do conteúdo: <i>(simulação, animação, site, etc).</i></p> <p>Animações e simulação de comportamento dos gases com diferentes densidades e em diferentes temperaturas.</p>
<p>Contexto: <i>(descrever em no máximo três linhas como deve acontecer esta atividade, se há materiais de suporte, se a sala necessita de um arranjo especial de cadeiras, se deve haver preparação de algum material).</i></p> <p>Trata de uma atividade de identificação de comportamento dos gases de acordo com suas densidades e com a variação de temperatura. Pode ser discutido com o parceiro se a atividade for elaborada em dupla. Esta atividade também pode ser desenvolvida individualmente.</p>
<p>Tempo previsto: <i>(indique o tempo previsto para esta atividade)</i></p> <p>Duas aulas de 50 minutos cada.</p>

**Método:**

Aluno: *(descrever o que o aluno faz na atividade passo a passo. Ex: 1- aluno clica no menu 2- aluno explora ambiente 3- aluno escreve as respostas).*

- 1- O aluno lê a primeira tela e interage com a primeira situação apresentada. Nesta situação ele deve perceber que gases diferentes possuem densidades diferentes. No final desta simulação o aluno responde à três questões levantadas pelo vendedor de balões para verificar se a simulação realmente foi compreendida.
- 2- O aluno passa para o segundo cenário, interagem com o mesmo. Neste cenário ele identifica a alteração do comportamento dos gases com a variação da temperatura.
- 3- No terceiro cenário o aluno participa de uma corrida de balões, onde aplica o conhecimento adquirido em outras etapas da atividade. Ao final da corrida há uma entrevista na qual o aluno responde às três perguntas feitas pela repórter.

Professor: *(descrever o que o professor faz na atividade passo a passo. Ex: 1- professor guia o alunos na atividade 2- professor observa aluno sem interferir 3- professor faz perguntas ao aluno após completar o passo 3).*

- 1- O professor apresenta uma pequena explicação sobre o conteúdo na sala de aula do que será abordado.
- 2- Levanta questões de discussão com alunos sobre a temática (guia do professor).
- 3- O professor inicia a aula na sala de computador deixando os alunos navegarem livremente na atividade.
- 4- Observa como ocorre a navegação.
- 5- Pede aos alunos um resumo do que ocorreu na utilização do objeto de aprendizagem na aula.
- 6- Pede aos alunos que discutam sobre suas dúvidas e o que acabaram de aprender.

Avaliação: *(descrever estratégias de avaliação).*

1. *Verifica a participação e interesse no uso do objeto de aprendizagem no computador e na etapa de discussão.*
2. *Verifica a ocorrência de dúvidas dos alunos a respeito do conteúdo aprendido.*
3. *Pode incluir a questão discutida no objeto em uma avaliação.*

Observações/Comentários:

ANEXO 2: Exemplos de OA de equipes RIVED



Tela do Objeto "Alimentos - Fonte de Energia"

Fonte: Equipe RIVED-UFMA

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rived name="Rived" >
  <cena id="0">
    <fala id="1" texto="Carboidratos, proteínas e lipídios"/>
  </cena>
  <cena id="1">
    <fala id="1" texto="Interessante a aula de hoje, vocês não acham?"/>
    <fala id="2" texto="É, realmente foi bom aprender sobre a química nos alimentos, tenho que cuidar mais da minha saúde!"/>
    <fala id="3" texto="Essa aula toda falando de comida me deu uma fome!!"/>
  </cena>
  <cena id="2">
    <fala id="1" texto="Qual o pedido?"/>
    <fala id="2" texto="Um sanduiche natural e um suco de manga."/>
    <fala id="3" texto="Um suco de acerola com adoçante."/>
    <fala id="4" texto="Um milkshake com batata fita e um X-TUDO."/>
    <fala id="5" texto="Nossa! Como você aguenta comer tudo isso?"/>
    <fala id="6" texto="Pois é amiga! Tô precisando fazer uma dieta..."/>
    <fala id="7" texto="Eu faço dieta."/>
    <fala id="8" texto="Fala sério!!!."/>
  </cena>
</rived>
```

Parte do arquivo XML do OA "Alimentos - Fonte de Energia"

Fonte: Equipe RIVED-UFMA

Tela do Objeto "Abundância"

Fonte: Equipe RIVED-UNESP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<tabela>
  <elemento tipo="hidrogenio,gasoso" id="Hidrogênio" simbolo="H" natomico="1">
    <isotopo massa="1" abundancia="99,9"></isotopo>
    <isotopo massa="2" abundancia="0,1"></isotopo>
    <isotopo massa="3" abundancia="*"></isotopo>
  </elemento>
  <elemento tipo="gases_nobres,gasoso" id="Hélio" simbolo="He" natomico="2">
    <isotopo massa="3" abundancia="0,000137"></isotopo>
    <isotopo massa="4" abundancia="99,999863"></isotopo>
  </elemento>
  <elemento tipo="metais_representativos_alcalinos,solido" id="Lítio" simbolo="Li" natomico="3">
    <isotopo massa="6" abundancia="7,59"></isotopo>
    <isotopo massa="7" abundancia="92,41"></isotopo>
  </elemento>
  <elemento tipo="metais_representativos_alcalinos-terrosos,solido" id="Berílio" simbolo="Be" natomico="4">
    <isotopo massa="9" abundancia="100"></isotopo>
  </elemento>
  <elemento tipo="semimetais,solido" id="Boro" simbolo="B" natomico="5">
    <isotopo massa="10" abundancia="19,9"></isotopo>
    <isotopo massa="11" abundancia="80,1"></isotopo>
  </elemento>
</tabela>
```

Parte do arquivo XML do OA "Abundância"

Fonte: Equipe RIVED-UNESP



Tela do Objeto "Espontaneidade"

Fonte: <http://www.rived.ufu.br/>

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<telai >
  <title> E utilizando um cano, é possível?</title>
  <textA> E se utilizarmos um cano, definindo o curso da água em todas as direções, a bomba consegue envia-la em sentido contrário ao curso do rio? </textA>
  <textA1> Clique na bomba para ver. </textA1>
  <textB> Olha só, a água está subindo pelo cano! Mas como isso está sendo possível?</textB>
</telai>
```

Parte do arquivo XML do OA "Espontaneidade"

Fonte: <http://www.rived.ufu.br/>



Tela do Objeto "Misturas"

Fonte: <http://sites.unifra.br/Portals/17/Quimica/Misturas>

ANEXO 3: XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <xs:import namespace="http://www.w3.org/2001/XMLSchema-instance"
schemaLocation="xsi.xsd"/>
  <xs:element name="object">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Rived"/>
      </xs:sequence>
      <xs:attribute ref="xsi:noNamespaceSchemaLocation" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Rived">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="LoMetadata"/>
        <xs:element ref="LOContents"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="LoMetadata">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="Anotation"/>
        <xs:element ref="Classification"/>
        <xs:element ref="Educational"/>
        <xs:element ref="General"/>
        <xs:element ref="LifeCycle"/>
        <xs:element ref="MetaMetadata"/>
        <xs:element ref="Relation"/>
        <xs:element ref="Rights"/>
        <xs:element ref="Technical"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="Anotation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="entity"/>
        <xs:element ref="date"/>
        <xs:element ref="description"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Classification">
    <xs:complexType>

```

```

    <xs:sequence>
      <xs:element ref="purpose"/>
      <xs:element ref="keyword"/>
      <xs:element ref="description"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="purpose">
  <xs:complexType/>
</xs:element>
<xs:element name="Educational">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="interactivityType"/>
      <xs:element ref="description"/>
      <xs:element ref="language"/>
      <xs:element ref="context"/>
      <xs:element ref="interactivityLevel"/>
      <xs:element ref="resource_type"/>
      <xs:element ref="typicalAgeRange"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="interactivityType">
  <xs:complexType/>
</xs:element>
<xs:element name="context">
  <xs:complexType/>
</xs:element>
<xs:element name="interactivityLevel">
  <xs:complexType/>
</xs:element>
<xs:element name="resource_type">
  <xs:complexType/>
</xs:element>
<xs:element name="typicalAgeRange">
  <xs:complexType/>
</xs:element>
<xs:element name="General">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="identifierCatalog"/>
      <xs:element ref="identifierEntry"/>
      <xs:element ref="title"/>
      <xs:element ref="language"/>
      <xs:element ref="description"/>
      <xs:element ref="keyword"/>
      <xs:element ref="discipline"/>
      <xs:element ref="module"/>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="discipline">
  <xs:complexType/>
</xs:element>
<xs:element name="module">
  <xs:complexType/>
</xs:element>
<xs:element name="LifeCycle">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="version"/>
      <xs:element ref="status"/>
      <xs:element ref="contributes"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="version">
  <xs:complexType/>
</xs:element>
<xs:element name="status">
  <xs:complexType/>
</xs:element>
<xs:element name="contributes">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="role"/>
      <xs:element ref="entity"/>
      <xs:element ref="date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="MetaMetadata">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="identifierCatalog"/>
      <xs:element ref="identifierEntry"/>
      <xs:element ref="schema"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="schema">
  <xs:complexType/>
</xs:element>
<xs:element name="Relation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="resource"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="resource">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element ref="kind"/>
    <xs:element ref="identifier_catalog"/>
    <xs:element ref="identifier_entry"/>
    <xs:element ref="description"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="kind">
  <xs:complexType/>
</xs:element>
<xs:element name="identifier_catalog">
  <xs:complexType/>
</xs:element>
<xs:element name="identifier_entry">
  <xs:complexType/>
</xs:element>
<xs:element name="Rights">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cost"/>
      <xs:element ref="copyright"/>
      <xs:element ref="description"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="cost">
  <xs:complexType/>
</xs:element>
<xs:element name="copyright">
  <xs:complexType/>
</xs:element>
<xs:element name="Technical">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="format"/>
      <xs:element ref="size"/>
      <xs:element ref="location"/>
      <xs:element ref="requirementType"/>
      <xs:element ref="requirementName"/>
      <xs:element ref="font"/>
      <xs:element ref="stage"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="format">
  <xs:complexType/>
</xs:element>
<xs:element name="location">
  <xs:complexType/>

```



```

</xs:element>
<xs:element name="requirementType">
  <xs:complexType/>
</xs:element>
<xs:element name="requirementName">
  <xs:complexType/>
</xs:element>
<xs:element name="stage">
  <xs:complexType/>
</xs:element>
<xs:element name="LOContents">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="CommonSceneElements"/>
      <xs:element ref="Scene"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="CommonSceneElements">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Button"/>
      <xs:element ref="Exercises"/>
      <xs:element ref="Text"/>
      <xs:element ref="ConceitualMap"/>
      <xs:element ref="Credits"/>
      <xs:element ref="Glosary"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="ConceitualMap">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="src"/>
      <xs:element ref="Button"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Credits">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="src"/>
      <xs:element ref="name"/>
      <xs:element ref="role"/>
      <xs:element ref="title"/>
      <xs:element ref="Text"/>
      <xs:element ref="Button"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Glosary">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element ref="title"/>
    <xs:element ref="term"/>
    <xs:element ref="Button"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="term">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="concept"/>
      <xs:element ref="definition"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="concept">
  <xs:complexType/>
</xs:element>
<xs:element name="definition">
  <xs:complexType/>
</xs:element>
<xs:element name="Scene">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Scenary"/>
    </xs:sequence>
    <xs:attribute name="frame" use="required"/>
    <xs:attribute name="id" use="required"/>
    <xs:attribute name="name" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Scenary">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Content"/>
    </xs:sequence>
    <xs:attribute name="id" use="required" type="xs:integer"/>
  </xs:complexType>
</xs:element>
<xs:element name="Content">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Text"/>
      <xs:element ref="image"/>
      <xs:element ref="Button"/>
      <xs:element ref="Table"/>
      <xs:element ref="Exercises"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="image">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="extension"/>
      <xs:element ref="path"/>
      <xs:element ref="width"/>
      <xs:element ref="height"/>
    </xs:sequence>
    <xs:attribute name="id" use="required"/>
    <xs:attribute name="posX" use="required"/>
    <xs:attribute name="posY" use="required"/>
    <xs:attribute name="sequence" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="extension">
  <xs:complexType/>
</xs:element>
<xs:element name="path">
  <xs:complexType/>
</xs:element>
<xs:element name="width">
  <xs:complexType/>
</xs:element>
<xs:element name="height">
  <xs:complexType/>
</xs:element>
<xs:element name="Table">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="tr"/>
    </xs:sequence>
    <xs:attribute name="id" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="tr">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="td"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="td">
  <xs:complexType/>
</xs:element>
<xs:element name="description">
  <xs:complexType/>
</xs:element>
<xs:element name="identifierCatalog">
  <xs:complexType/>
</xs:element>
<xs:element name="identifierEntry">

```

```

    <xs:complexType/>
</xs:element>
<xs:element name="title">
  <xs:complexType/>
</xs:element>
<xs:element name="language">
  <xs:complexType/>
</xs:element>
<xs:element name="keyword">
  <xs:complexType/>
</xs:element>
<xs:element name="entity">
  <xs:complexType/>
</xs:element>
<xs:element name="date">
  <xs:complexType/>
</xs:element>
<xs:element name="size">
  <xs:complexType/>
</xs:element>
<xs:element name="font">
  <xs:complexType/>
</xs:element>
<xs:element name="role">
  <xs:complexType/>
</xs:element>
<xs:element name="Text">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="text"/>
      <xs:element ref="font"/>
      <xs:element ref="size"/>
      <xs:element ref="category"/>
    </xs:sequence>
    <xs:attribute name="id" use="required"/>
    <xs:attribute name="posX" use="required"/>
    <xs:attribute name="posY" use="required"/>
    <xs:attribute name="sequence" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="category">
  <xs:complexType/>
</xs:element>
<xs:element name="Button">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="name"/>
      <xs:element ref="text"/>
      <xs:element ref="action"/>
      <xs:element ref="help"/>
      <xs:element ref="type"/>
    </xs:choice>
  </xs:complexType>

```

```

    </xs:choice>
    <xs:attribute name="id" use="required"/>
    <xs:attribute name="posX" use="required"/>
    <xs:attribute name="posY" use="required"/>
    <xs:attribute name="sequence" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="action">
  <xs:complexType/>
</xs:element>
<xs:element name="help">
  <xs:complexType/>
</xs:element>
<xs:element name="type">
  <xs:complexType/>
</xs:element>
<xs:element name="Exercises">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Question"/>
      <xs:element ref="feedback"/>
      <xs:element ref="Button"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Question">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="text"/>
      <xs:element ref="answer"/>
      <xs:element ref="correctAnswer"/>
    </xs:sequence>
    <xs:attribute name="id" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="answer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="text"/>
      <xs:element ref="feedback"/>
    </xs:sequence>
    <xs:attribute name="id" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="correctAnswer">
  <xs:complexType/>
</xs:element>
<xs:element name="src">
  <xs:complexType/>
</xs:element>
<xs:element name="name">

```

```
<xs:complexType/>
</xs:element>
<xs:element name="text">
  <xs:complexType/>
</xs:element>
<xs:element name="feedback">
  <xs:complexType>
    <xs:attribute name="id"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

ANEXO 4: Estrutura de informação padrão em XML

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<object xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="schema_altova.xsd">
  <Rived>
    <LoMetadata> {Metadados LOM}
      <General>
        <identifierCatalog></identifierCatalog>
        <identifierEntry> </identifierEntry>
        <title> </title>
        <language> </language>
        <description> </description>
        <keyword> </keyword>
        <discipline> </discipline>
        <module> </module>
      </General>
      <LifeCycle>
        <version> </version>
        <status> </status>
        <contributes>
          <role> </role>
          <entity> </entity>
          <date/>
        </contributes>
      </LifeCycle>
      <MetaMetadata>
        <identifierCatalog> </identifierCatalog>
        <identifierEntry> </identifierEntry>
        <schema> </schema>
      </MetaMetadata>
      <Technical>
        <format> </format>
        <size> </size>
        <location> </location>
        <requirementType> </requirementType>
        <requirementName> </requirementName>
        <font> </font>
        <stage> </stage>
      </Technical>
      <Educational>
        <interactivityType> </interactivityType>
        <description> </description>
        <language> </language>
        <context> </context>
        <interactivityLevel> </interactivityLevel>
        <resource_type> </resource_type>
        <typicalAgeRange> </typicalAgeRange>
      </Educational>
    </LoMetadata>
  </Rived>
</object>

```

```

        <cost> </cost>
        <copyright> </copyright>
        <description> </description>
    </Rights>
    <Relation>
        <resource>
            <kind> </kind>
            <identifier_catalog> </identifier_catalog>
            <identifier_entry> </identifier_entry>
            <description> </description>
        </resource>
    </Relation>
    <Anotation>
        <entity> </entity>
        <date> </date>
        <description> </description>
    </Anotation>
    <Classification>
        <purpose> </purpose>
        <keyword> </keyword>
        <description> </description>
    </Classification>
</LoMetadata>
<LOContents> {Informações de cenas e cenários}
    <CommonSceneElements> { Informações gerais }
        <Text id="" sequence="" posX="" posY="">
            <text> </text>
            <font> </font>
            <size/>
            <category> </category>
        </Text>
        <ConceitualMap>
            <src/>
            <Button id="" sequence="" posX="" posY="">
                <name> </name>
                <text> </text>
                <help> </help>
                <action> </action>
                <type> </type>
            </Button>
        </ConceitualMap>
    <Glosary>
        <title/>
        <term>
            <concept/>
            <definition/>
        </term>
        <Button id="" sequence="" posX="" posY="">
            <name> </name>
            <text> </text>
            <help> </help>

```



```

        <action/>
        <type> </type>
    </Button>
</Glosary>
<Exercises>
    <Question id="">
        <text> </text>
        <answer id="">
            <text/>
            <feedback/>
        </answer>
        <correctAnswer> </correctAnswer>
    </Question>
    <feedback id=""> </feedback>
    <Button id="" sequence="" posX="" posY="">
        <name/>
        <text/>
        <help/>
        <action/>
        <type/>
    </Button>
</Exercises>
<Credits>
    <src/>
    <name> </name>
    <role> </role>
    <title> </title>
    <Text id="" sequence="" posX="" posY="">
        <text/>
        <font> </font>
        <size/>
        <category/>
    </Text>
    <Button id="" sequence="" posX="" posY="">
        <name/>
        <text/>
        <help/>
        <action/>
        <type/>
    </Button>
</Credits>
<Button id="" sequence="" posX="" posY="">
    <name> </name>
    <text> </text>
    <help> </help>
    <action/>
    <type> </type>
</Button>
</CommonSceneElements>
<Scene id="" name=" " frame="">
    <Scenary id="1">

```

```

<Content>
  <Text id="" sequence="" posX="" posY="">
    <text> </text>
    <font> </font>
    <size> </size>
    <category> </category>
  </Text>
  <image id="" sequence="" posX="" posY="">
    <extension/>
    <path> </path>
    <width> </width>
    <height> </height>
  </image>
  <Button id="" sequence="" posX="" posY="">
    <name> </name>
    <text/>
    <help/>
    <action/>
    <type/>
  </Button>
  <Table id="">
    <tr>
      <td/>
    </tr>
  </Table>
  <Exercises>
    <Question id="">
      <text> </text>
      <answer id="">
        <text/>
        <feedback/>
      </answer>
      <correctAnswer> </correctAnswer>
    </Question>
    <feedback id=""> </feedback>
    <Button id="" sequence="" posX="" posY="">
      <name/>
      <text/>
      <help/>
      <action/>
      <type/>
    </Button>
  </Exercises>
</Content>
</Scenary>
</Scene>
</LOContents>
</Rived>
</object>

```

ANEXO 5: Cena do OA Balão na estrutura padrão

```

<Scene id="0" name="Inicial" frame=""> {1ª CENA- INICIAL}
  <Scenary id="1">
    <Content>
      <Text id="1" sequence="1" posX="" posY="">
        <text>Neste menu você encontra os links para o início do
módulo, os exercícios de fixação, o mapa de conceitos do objeto, o glossário e os
créditos.</text>
        <font>Arial</font>
        <size> 12 </size>
        <category>explicativo </category>
      </Text>
      <Text id="2" sequence="2" posX="" posY="">
        <text>Este é o menu de navegação de cenários. Cenários
são as diferentes situações que você vai encontrar no objeto. Os números indicam a
quantidade de cenários. Aqui você pode escolher por onde quer começar a sua
navegação.</text>
        <font>Arial</font>
        <size> 12 </size>
        <category> explicativo</category>
      </Text>
      <Text id="3" sequence="3" posX="" posY="">
        <text>Aqui ficam os botões de navegação pelas telas.
Utilize a seta para a direita para avançar e a seta para esquerda para retornar à tela
anterior. O botão central leva à tela inicial.</text>
        <font>Arial</font>
        <size> 12 </size>
        <category> explicativo</category>
      </Text>
      <image id="1" sequence="1" posX="" posY="">
        <extension>fundoParque.bmp</extension>
        <path> </path>
        <width> 600</width>
        <height> 400</height>
      </image>
      <image id="2" sequence="2" posX="" posY="">
        <extension>menino.bmp</extension>
        <path> </path>
        <width> </width>
        <height> </height>
      </image>
      <image id="3" sequence="3" posX="" posY="">
        <extension>explicação.bmp</extension>
        <path> </path>
        <width> </width>
        <height> </height>
      </image>
      <Button id="1" sequence="" posX="" posY="">
        <name>Entrar </name>

```



```
<Button id="1" sequence="" posX="" posY="">
  <name>avançar</name>
  <text> </text>
  <help>Avançar</help>
  <action>Avançar para o próximo cenário</action>
  <type>navegacao</type>
</Button>
<Button id="2" sequence="" posX="" posY="">
  <name>retornar</name>
  <text> </text>
  <help>Retornar</help>
  <action>Voltar para cenário anterior</action>
  <type>navegacao</type>
</Button>
<Button id="3" sequence="" posX="" posY="">
  <name>Exibir</name>
  <text> </text>
  <help>Exibir explicação</help>
  <action>exibir explicação</action>
  <type>explicativo</type>
</Button>
<Button id="4" sequence="" posX="" posY="">
  <name>Ocultar</name>
  <text> </text>
  <help>Ocultar explicação</help>
  <action>ocultar a explicação</action>
  <type>explicativo</type>
</Button>
</Content>
</Scenary>
</Scene>
```

ANEXO 6: Questionário de avaliação de editores XML**QUESTIONÁRIO DE AVALIAÇÃO DE EDITORES XML**

AVALIADOR:
SOFTWARE AVALIADO:

DATA: __/__/__

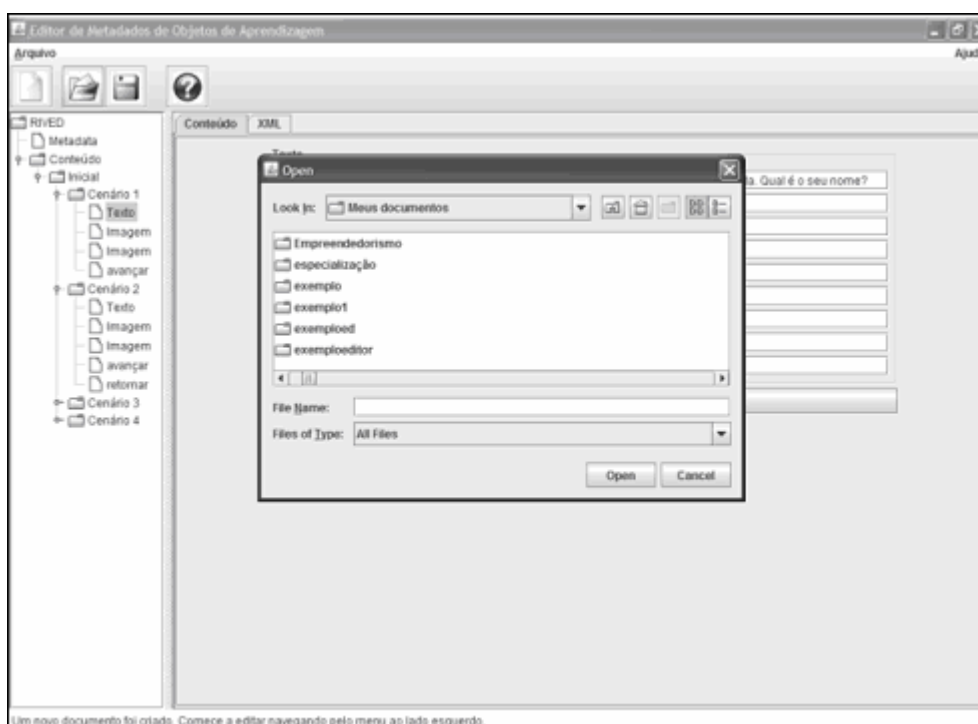
LEGENDA	
Símbolo	Significado
S	Sim
P+	Parcialmente com POUCAS restrições
P	Parcialmente
P-	Parcialmente com MUITAS restrições
N	Não

Responda as questões a seguir, conforme indicação da legenda.

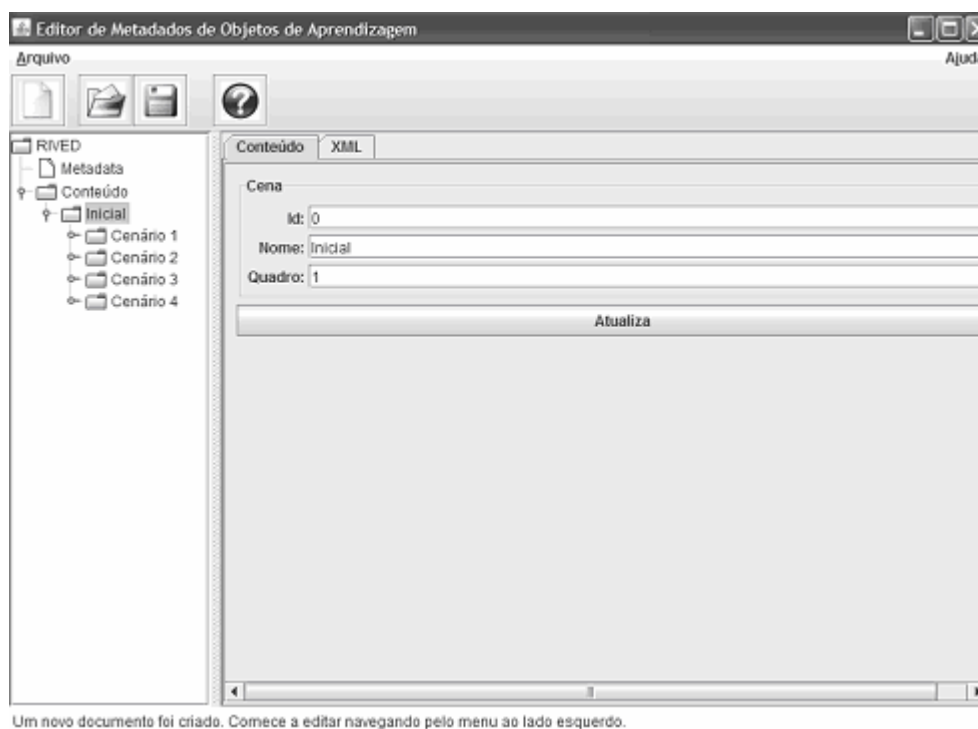
A - Critérios baseados na ISO 9126 – Funcionalidade	S	P+	P	P-	N
A1- O <i>software</i> faz o que é apropriado.					
A2- O <i>software</i> faz de forma correta o que foi proposto.					
B - Critérios baseados na ISO 9126 – Usabilidade	S	P+	P	P-	N
B1- O <i>software</i> é de fácil aprendizado.					
B2- As representações das funções de interface (botões, ícones, <i>helps</i> , etc.) são fáceis de serem entendidas.					
B3- O editor é autoajustável de acordo com o tamanho do monitor, permitindo boa visibilidade.					
B4- Os <i>helps</i> dos botões (ativados quando o cursor do mouse está sobre os botões) auxiliam quanto ao uso.					

B5- O sistema de ajuda é adequado, ou seja, explica as dúvidas do usuário adequadamente.					
B6- Os títulos de menus são satisfatoriamente explicativos, refletindo a natureza da escolha a ser feita.					
B7- Os textos existentes entre cada <i>tag</i> de abertura e fechamento aparecem de forma organizada.					
B8- O <i>software</i> é de fácil operação e controle.					
C - Critérios baseados na elicitação de requisitos para um editor de OA	S	P +	P	P -	N
C1- O <i>software</i> exibe as tags de abertura e fechamento (<> </>).					
C2- O <i>software</i> permite renomear o conteúdo das <i>tags</i> .					
C3- É complexo criar e alterar o arquivo XML com o <i>software</i> .					
C4- O <i>software</i> oferece/permite uma padronização (utiliza um padrão) para a estrutura de informação dos OA.					
C5- O <i>software</i> permite inserir <i>tags</i> e seu conteúdo .					
C6- O <i>software</i> permite remover <i>tags</i> e seu conteúdo.					
C7- O <i>software</i> permite a validação do arquivo XML.					
C8- O <i>software</i> parece adequado para a criação de arquivos XML para OA.					

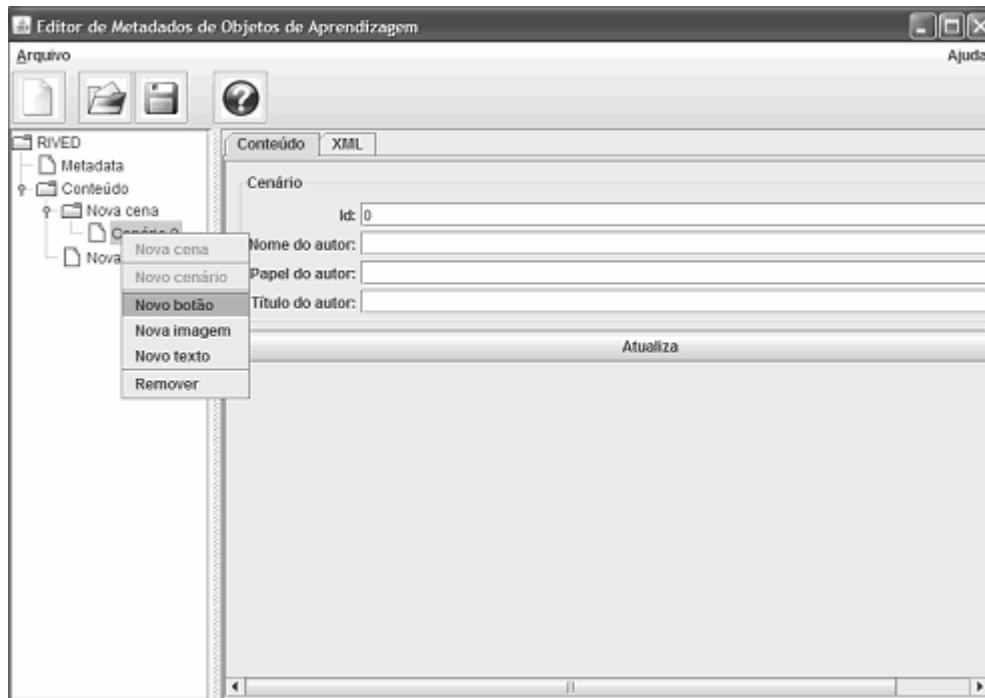
ANEXO 7: Telas da ferramenta correspondentes a cada caso de uso



Tela que representa o caso de uso Abrir Arquivo XML

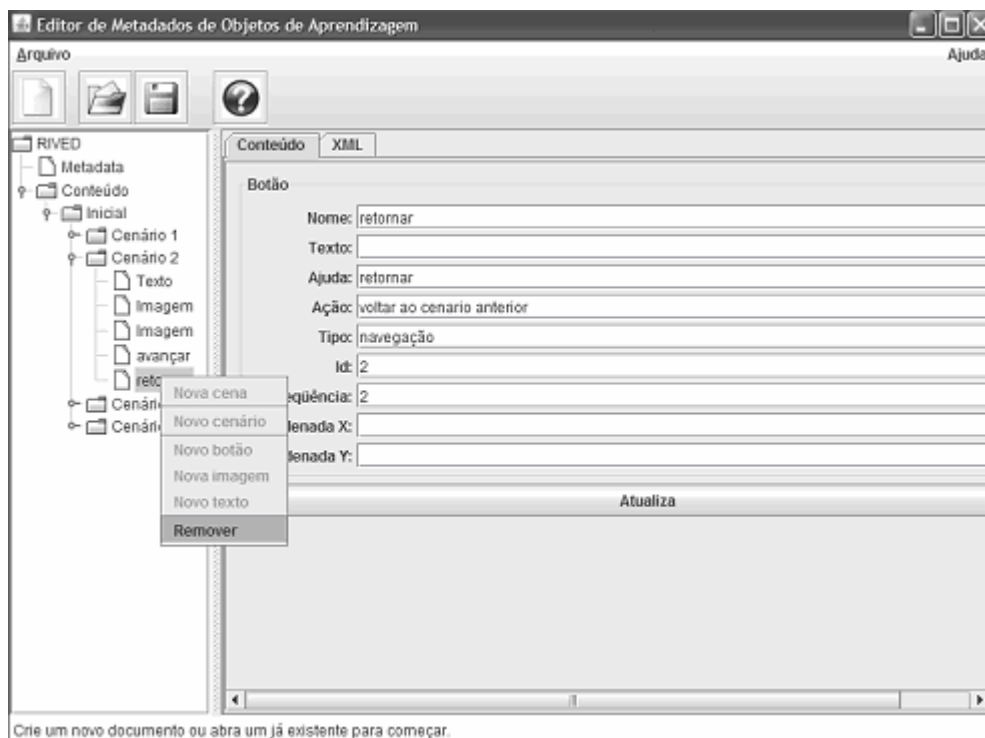


Tela que representa o caso de uso Modificar Conteúdo do Arquivo XML



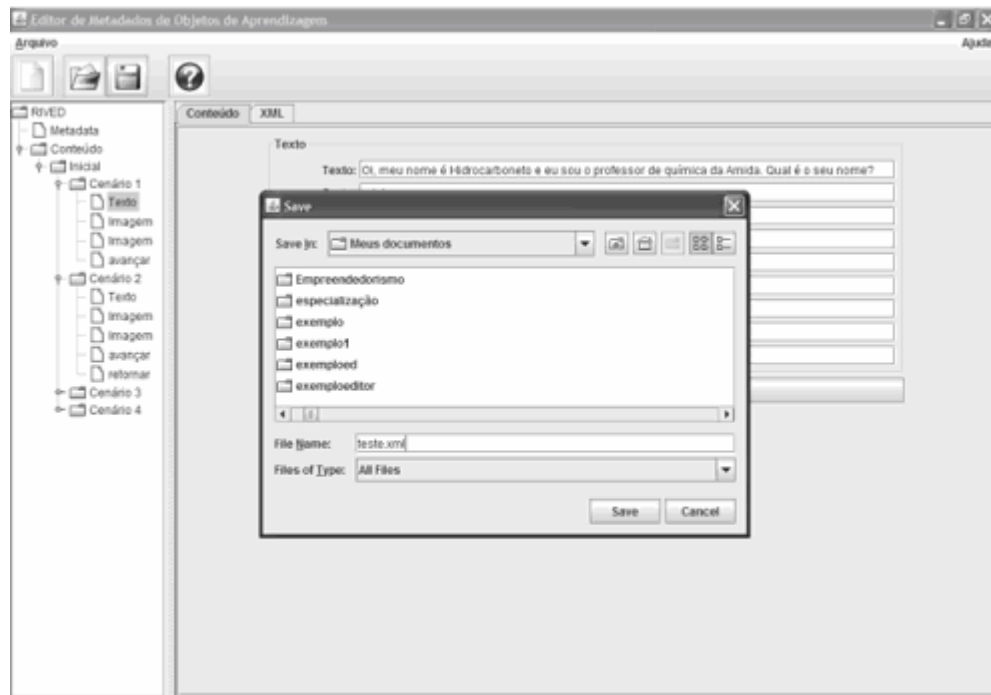
Um novo documento foi criado. Comece a editar navegando pelo menu ao lado esquerdo.

Tela que representa o caso de uso Inserir Elemento

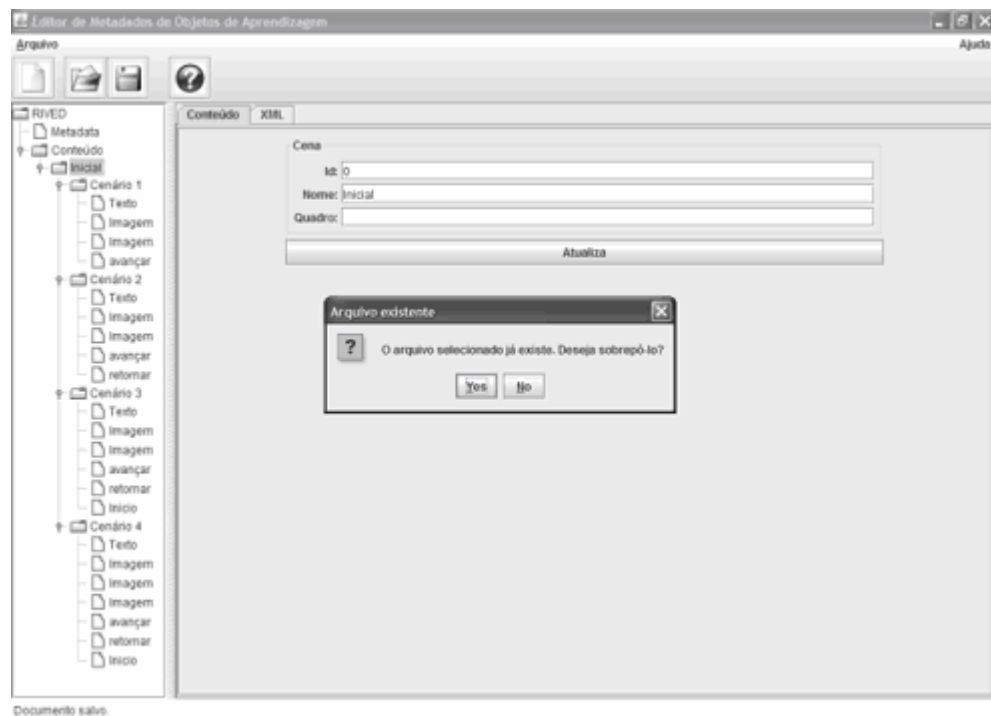


Crie um novo documento ou abra um já existente para começar.

Tela que representa o caso de uso Remover elemento



Tela que representa o caso de uso Salvar Arquivo XML



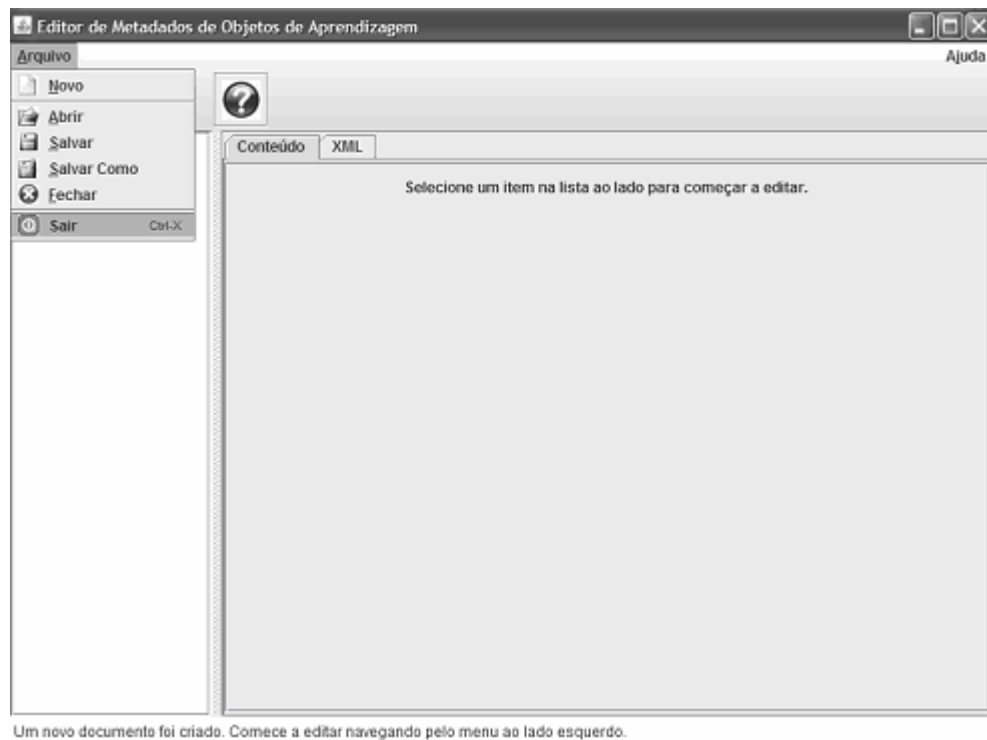
Tela que representa o caso de uso Salvar Como - Arquivo XML



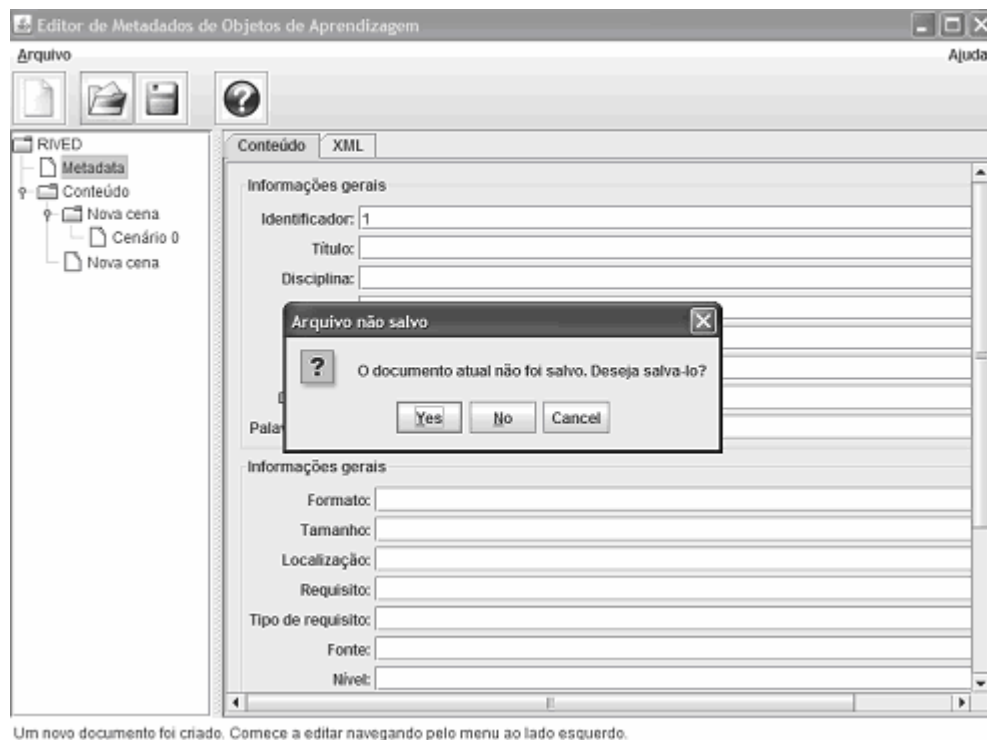
Tela que representa o caso de uso Abrir Novo



Tela que representa o caso de uso Fechar arquivo



Tela que representa o caso de uso Sair



Tela que representa a opção Salvar antes dos casos de uso Abrir Novo, Fechar e Sair

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)