

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ALEXANDRO BORDIGNON

**Proposta para Criação e Catalogação de  
Objetos de Aprendizagem Interoperáveis**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de Mestre em Ciência  
da Computação

Profa. Dra. Rosa Maria Vicari  
Orientador

Prof. Dr. Valter Roesler  
Co-orientador

Porto Alegre, dezembro de 2010.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Bordignon, Alexandro

Proposta para Criação e Catalogação de Objetos de Aprendizagem Interoperáveis / Alexandro Bordignon – Porto Alegre: Programa de Pós-Graduação em Computação, 2010.

124 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2010. Orientador: Profa. Dra. Rosa Maria Vicari; Co-orientador: Valter Roesler.

1.TV digital. 2.Objetos de Aprendizagem 3.Interoperabilidade 4.Metadados. I. Vicari, Rosa Maria. II. Roesler, Valter. III. Proposta para Criação e Catalogação de Objetos de Aprendizagem Interoperáveis.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

Agradeço às pessoas que me auxiliaram, incentivaram e que contribuíram para a realização deste trabalho.

Primeiramente a minha esposa, Fabiane, pelo amor, paciência, compreensão e apoio incondicional, me dando força mesmo quando necessitava direcionar minha atenção para o mestrado. Amo você!

Aos meus pais, Oneida e Nestor, exemplos de humildade e persistência, base essencial para conquista de meus objetivos e, especialmente, pela compreensão nos momentos de ausência. À minha irmã, pelo incentivo e torcida em todos os momentos da minha vida. Amo vocês.

Aos meus orientadores, Profa. Dra. Rosa Maria Vicari e Prof. Dr. Valter Roesler, pelo exemplo, ajuda, incentivo e, principalmente, pela oportunidade de aprendizado ao trabalhar com pessoas extremamente competentes, cujas lições estarão comigo não apenas no mestrado, mas por toda vida. Admiro muito vocês.

Aos meus colegas do laboratório PRAV e do projeto OBAA, em especial ao Elder, Tiago, Fernando, Guilherme, Julio, Malu, Ewerton, André e Adriano, os quais contribuíram imensamente para este trabalho.

A todos meus amigos, que sempre que me encontravam queriam saber como estava indo o mestrado.

Ao Instituto de Informática da UFRGS e seus professores, pela qualidade de ensino e pela oportunidade que me foi dada de estudar em um ambiente tão rico, do qual me orgulho de fazer parte.

As dificuldades e o meu crescimento durante o mestrado foram acompanhados por muitas pessoas especiais para mim e, a todas elas, eu agradeço cada segundo que estiveram ao meu lado durante nesse período.

A todos, novamente o meu muito obrigado. Sou imensamente grato.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>7</b>
<b>LISTA DE FIGURAS.....</b>	<b>11</b>
<b>LISTA DE TABELAS.....</b>	<b>13</b>
<b>RESUMO .....</b>	<b>14</b>
<b>ABSTRACT .....</b>	<b>15</b>
<b>1 INTRODUÇÃO .....</b>	<b>16</b>
1.1 Objetivos .....	17
1.2 Trabalhos Relacionados.....	18
1.3 Outras Soluções para Interoperabilidade .....	20
1.4 Estrutura do Trabalho.....	22
<b>2 PADRÕES INERENTES ÀS PLATAFORMAS.....</b>	<b>23</b>
2.1 Computadores Pessoais.....	23
2.1.1 Áudio e Vídeo.....	24
2.1.2 Imagens.....	24
2.1.3 Extensible HyperText Markup Language (XHTML) .....	25
2.1.4 Cascading Style Sheet (CSS).....	26
2.1.5 Linguagem <i>Script</i> .....	27
2.1.6 Java .....	27
2.1.7 Flash.....	28
2.1.8 HTML5 .....	28
2.1.9 Usabilidade na Web.....	29
2.2 TV Digital.....	29
2.2.1 Interatividade na TV Digital .....	30
2.2.2 O <i>Middleware</i> do SBTVD e Ambientes Declarativo e Procedural.....	33
2.2.3 Áudio e Vídeo.....	33
2.2.4 Imagens.....	34
2.2.5 Ambiente Declarativo .....	34
2.2.6 Linguagem <i>Script</i> .....	35

2.2.7	Java .....	36
2.2.8	Usabilidade na TV Digital .....	36
<b>2.3</b>	<b>Dispositivos Móveis .....</b>	<b>39</b>
2.3.1	Áudio e Vídeo .....	40
2.3.2	Imagens .....	41
2.3.3	Extensible HyperText Markup Language (XHTML) .....	42
2.3.4	Cascating Style Sheet (CSS) .....	42
2.3.5	Linguagem <i>Script</i> .....	43
2.3.6	Java .....	43
2.3.7	Flash .....	43
2.3.8	Usabilidade em Dispositivos Móveis .....	44
<b>3</b>	<b>PADRÕES DE METADADOS .....</b>	<b>45</b>
3.1	<i>Dublin Core Metadata Initiative (DCMI)</i> .....	45
3.2	MPEG-7 .....	46
3.3	MPEG-21 .....	49
3.4	IEEE Learning Object Metadata (IEEE-LOM) .....	50
3.5	SCORM .....	51
3.6	Tabelas SI/PSI .....	52
3.7	TV-ANYTIME .....	55
3.7.1	Arquitetura do TV-Anytime em um Sistema de <i>Broadcast</i> .....	57
3.7.2	Modelagem de Metadados do TV-Anytime .....	58
3.8	MXF ( <i>Material eXchange Format</i> ) .....	62
3.8.1	Metadados do Arquivo MXF .....	64
<b>4</b>	<b>PROPOSTAS PARA INTEROPERABILIDADE .....</b>	<b>66</b>
4.1	<b>Criação de Objetos de Aprendizagem Interoperáveis .....</b>	<b>66</b>
4.1.1	Interface com o usuário .....	67
4.1.2	Recomendações para Conteúdo Texto .....	67
4.1.3	Imagens .....	68
4.1.4	Vídeo .....	68
4.1.5	Áudio .....	69
4.1.6	Outros formatos de mídia .....	69
4.2	<b>Mecanismos de adaptabilidade .....</b>	<b>69</b>
4.2.1	Módulo Servidor .....	70
4.2.2	Módulo <i>TagLibs</i> .....	71
4.2.3	<i>Website Downloader</i> .....	74
4.2.4	<i>Website Converter</i> .....	75
4.3	<b>Conjuntos de Metadados de Extensão aos Padrões Educacionais .....</b>	<b>76</b>
4.3.1	Metadados de Adaptação .....	77
4.3.2	Metadados para Segmentação .....	79
<b>5</b>	<b>IMPLEMENTAÇÕES PARA VALIDAR A PROPOSTA .....</b>	<b>81</b>

5.1	Metodologia para Escolha dos Objetos de Aprendizagem.....	81
5.2	Ambiente de Validação e Ferramentas de Conversão de Formatos .....	83
5.3	Conversão do Objeto de Aprendizagem “De onde vem a televisão?” .....	84
5.4	Conversão do Objeto “Outras Infâncias” .....	88
5.5	Implementação do Curso Educacional “Viva Saudável” .....	92
5.6	Conclusões sobre os Objetos de Aprendizagem Desenvolvidos .....	96
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>98</b>
	<b>APÊNDICE A – TERMINAIS DE TV DIGITAL E POSSÍVEIS CENÁRIOS EDUCACIONAIS.....</b>	<b>101</b>
A.1	Configuração 1 – Terminal de Acesso Básico.....	101
A.2	Configuração 2 – Terminal de Acesso com Recurso de Armazenamento em Massa .....	102
A.3	Configuração 3 – Terminal de Acesso com <i>Middleware</i> e Sem Canal de Interatividade ....	102
A.4	Configuração 4 – Terminal de Acesso com <i>Middleware</i> e com Canal de Interatividade ....	103
A.5	Configuração 5 – Terminal de Acesso com Canal de Interatividade e sem <i>Middleware</i> .....	103
	<b>APÊNDICE B – METADADOS TÉCNICOS PROPOSTOS .....</b>	<b>104</b>
	<b>APÊNDICE C – METADADOS DE SEGMENTAÇÃO PROPOSTOS .....</b>	<b>108</b>
	<b>APÊNDICE D – FERRAMENTAS UTILIZADAS.....</b>	<b>111</b>
D.1	Ferramentas para Conversão de Formatos .....	111
D.2	Ferramentas para Compor o Servidor de Aplicação.....	113
D.2.1	Ferramentas de Streaming .....	114
D.2.2	Servidor Web .....	116
	<b>REFERÊNCIAS.....</b>	<b>117</b>

## LISTA DE ABREVIATURAS E SIGLAS

AAC	<i>Advanced Audio Coding</i>
AAF	<i>Advanced Authoring Format</i>
ABNT	Associação Brasileira de Normas Técnicas
AMR	<i>Adaptive Multi-Rate Codec</i>
ASF	<i>Active Streaming Format</i>
ARIB	<i>Association of Radio Industries and Businesses</i>
ATSC	<i>Advanced Television System Committee</i>
AVC	<i>Advanced Video Coding</i>
BBC	<i>British Broadcasting Corporation</i>
BIOE	Banco Internacional de Objetos Educacionais
BNDIGITAL	Biblioteca Nacional Digital Brasil
CAT	<i>Conditional Access Table</i>
CDC	<i>Connected Device Configuration</i>
CESTA	Coletânea de Entidades de Suporte ao Uso de Tecnologia na Aprendizagem
CINTED	Centro Interdisciplinar de Novas Tecnologias na Educação
CLDC	<i>Connected Limited Device Configuration</i>
CRID	<i>Content Reference Identification</i>
CSS	<i>Cascading Style Sheet</i>
DAVIC	<i>Digital Audio Video Council</i>
DCMI	<i>Dublin Core Metadata Initiative</i>
DDL	<i>Description Definition Language</i>
DIA	<i>Digital Item Adaptation</i>
DID	<i>Digital Item Declaration</i>
DII	<i>Digital Item Identification</i>
DS	<i>Description Schemes</i>
DSM-CC	<i>Digital Storage Media, Command and Control</i>

DTV	<i>Digital Television</i>
DVB	<i>Digital Video Broadcasting</i>
ECMA	<i>European Computer Manufacturers Association</i>
EMM	<i>Entitlement Management Messages</i>
ERT	<i>Event Relation Table</i>
ESMP	<i>ECMA Script - Mobile Profile</i>
EPG	<i>Electronic Program Guide</i>
ES	<i>Elementary Stream</i>
FEB	<i>Federação de Repositórios Educa Brasil</i>
GEM	<i>Globally Executable MHP</i>
GIF	<i>Graphics Interchange Format</i>
HAVi	<i>Home Audio Video Interoperability</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IPMP	<i>Intellectual Property Management and Protection Components</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
ISDB-T	<i>Integrated Services Digital Broadcasting - Terrestrial</i>
ISDB-Tb	<i>ISDB-T - Brazilian Version</i>
ITU	<i>International Telecommunication Union</i>
JavaEE	<i>Java Platform, Enterprise Edition</i>
JavaME	<i>Java Platform, Micro Edition</i>
JavaSE	<i>Java Platform, Standard Edition</i>
JPEG	<i>Joint Photographic Experts Group</i>
JSF	<i>Java Server Faces</i>
JSP	<i>Java Server Pages</i>
JVM	<i>Java Virtual Machine</i>
KLV	<i>Key Length Value</i>
LIT	<i>Local Event Table</i>
LMS	<i>Learning Management Systems</i>
LOM	<i>Learning Object Metadata</i>
METS	<i>Metadata Encoding and Transmission Standard</i>
MHP	<i>Multimedia Home Platform</i>
MIDP	<i>Mobile Information Device Profile</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MOV	<i>Movie</i>



MP3	<i>MPEG Audio Layer 3</i>
MPEG	<i>Moving Picture Experts Group</i>
MXF	<i>Material eXchange Format</i>
NCL	<i>Nested Context Language</i>
NDR	<i>Network Digital Recorder</i>
NUTED	Núcleo de Tecnologia Digital Aplicada à Educação
OA	Objeto de Aprendizagem
OBAA	Objetos de Aprendizagem Baseados em Agentes
OMA	<i>Open Mobile Alliance</i>
PAT	<i>Program Association Table</i>
PC	<i>Personal Computers</i>
PCM	<i>Pulse-code Modulation</i>
PDA	<i>Personal Digital Assistant</i>
PDR	<i>Personal Digital Recorder</i>
PES	<i>Packetized Elementary Stream Packets</i>
PID	<i>Packet Identifier</i>
PMT	<i>Program Map Table</i>
PNG	<i>Portable Network Graphics</i>
PRAV	Projetos em Áudio e Vídeo
PS	<i>Parametric Stereo</i>
PSI	<i>Program Specific Information</i>
PVR	<i>Personal Video Recorder</i>
QVGA	<i>Quarter Video Graphics Array</i>
RA	<i>Real Audio</i>
RDD	<i>Rights Data Dictionary</i>
REL	<i>Rights Expression Language</i>
RIA	<i>Rich Web Applications</i>
RM	<i>Real Media</i>
SAVANT	<i>Synchronised and Scalable AV content Across NeTworks</i>
SBR	<i>Spectral Band Replication</i>
SBTVD	Sistema Brasileiro de Televisão Digital
SCORM	<i>Sharable Content Object Reference Model</i>
SI	<i>Service Information</i>
SMPTE	<i>Society of Motion Picture and Television Engineers</i>
SUPER	<i>Simplified Universal Player Encoder &amp; Renderer</i>

SVG	<i>Scalable Vector Graphics</i>
TDT	<i>Time and Date Table</i>
TIFF	<i>Tagged Image File Format</i>
TOT	<i>Time Offset Table</i>
TS	<i>Transport Stream</i>
TV	Televisão
TVA	<i>TV-Anytime</i>
TVD	Televisão Digital
UFRGS	Universidade Federal do Rio Grande do Sul
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
WAE	<i>Wireless Application Environment</i>
WBMP	<i>Wireless Bitmap</i>
WMA	<i>Windows Media Audio</i>
WMV	<i>Windows Media Video</i>
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>Extensible Markup Language</i>

## LISTA DE FIGURAS

Figura 1. Recomendação de infra-estrutura interoperável para Web. ....	19
Figura 2. Exemplo de tag HTML para criação de link entre páginas.....	25
Figura 3. Exemplo de Applet em uma página Web (a) e estrutura do código Java do Applet (b) .....	28
Figura 4. Carrossel de dados na TV digital.....	31
Figura 5. Transmissão e recepção de sinal de TV digital terrestre .....	32
Figura 6. Arquitetura do middleware Ginga.....	33
Figura 7. Famílias de tipos utilizados pelas emissoras BBCi e SKY .....	37
Figura 8. Tamanhos de textos sugeridos para a interface .....	38
Figura 9. Utilização de botões coloridos .....	39
Figura 10. Hierarquia de uma pesquisa no TGN para “Porto Alegre” .....	46
Figura 11. Elementos Principais das Descrições MPEG-7 .....	47
Figura 12. Exemplo de cadeia MPEG-7 .....	48
Figura 13. Partes relacionadas ao MPEG-21 .....	49
Figura 14. Fluxos elementares formando um feixe de transporte MPEG-2 .....	52
Figura 15. Exemplo de um MPEG-2 Transport Stream.....	53
Figura 16. Exemplo de um EPG.....	53
Figura 17. Serviços descritos através de conjunto de tabelas PSI.....	54
Figura 18. Relacionamento entre a Transport Stream e as tabelas SI.....	54
Figura 19. Sistema de broadcast.....	57
Figura 20. Documentos TV-Anytime com “TVAMain” como elemento raiz .....	59
Figura 21. Componentes principais de um arquivo MXF simples .....	63
Figura 22. Áreas de metadados em MXF .....	64
Figura 23. Elementos do cabeçalho HTTP, de uma requisição do Firefox 3.0.5. ....	71
Figura 24. Exemplo de uso de uma taglib fictícia.....	72
Figura 25. Exemplo de geração condicional de código XHTML.....	74
Figura 26. Funcionamento do Website Downloader.....	75
Figura 27. Tela inicial Website Downloader .....	75
Figura 28. Tela inicial do Site Adapter.....	76
Figura 29. Trecho do código fonte do objeto de aprendizagem “De onde vem a televisão?” .....	85
Figura 30. Código XHTML gerado pelo Interop a partir do acesso por PCs.....	85
Figura 31. Código XHTML gerado pelo Interop a partir do acesso por dispositivos móveis .....	85
Figura 32. Código NCL para execução do OA na TV digital. ....	86
Figura 33. Objeto de Aprendizagem “De Onde vem a TV?”, apresentado nas três plataformas.....	86
Figura 34. XML dos metadados de extensão ao grupo técnico.....	87
Figura 35. XML com dois segmentos do vídeo “De onde vem a televisão?” .....	88
Figura 36. Objeto de aprendizagem “Outras Infâncias“ no formato original .....	89
Figura 37. Trecho de código fonte do objeto de aprendizagem “Outras Infâncias” .....	90
Figura 38. Objeto de Aprendizagem “Outras Infâncias” nas plataformas Web, TV digital e móvel .....	90
Figura 39. Trecho NCL onde os arquivos HTML são referenciados .....	91
Figura 40. XML com metadados técnicos do OA.....	91
Figura 41. Exemplo dos metadados de segmentação.....	92
Figura 42. Página inicial do curso na Web, celular e TV digital .....	93
Figura 43. Exemplo de página de conteúdo do objeto de aprendizagem “Viva Saudável”.....	94
Figura 44. Exercícios interativos no Viva Saudável .....	94
Figura 45. XML com metadados técnicos do OA “Viva Saudável”.....	95
Figura 46. Metadados de segmentação do OA “Viva Saudável” .....	95

<i>Figura 47. Ferramenta SUPER, utilizada de conversão de formatos de áudio e vídeo.....</i>	<i>112</i>
<i>Figura 48. Interface do VLC Media Player.....</i>	<i>113</i>
<i>Figura 49. Conversão de mídias no VLC Media Player. ....</i>	<i>113</i>
<i>Figura 50. Página de configuração do DSS.....</i>	<i>115</i>

## LISTA DE TABELAS

<i>Tabela 2.1: Codificação de vídeo no sistema brasileiro de TV digital.....</i>	<i>34</i>
<i>Tabela 2.2: Definições do Contexto Padrão de Entrega.....</i>	<i>42</i>
<i>Tabela 3.1: Categorias de Metadados do TV-Anytime.....</i>	<i>59</i>
<i>Tabela 3.2. Conjuntos de metadados DMS-1 e seu uso.....</i>	<i>64</i>
<i>Tabela 3.3. Metadados de Propriedades do Grupo Titles.....</i>	<i>65</i>
<i>Tabela 4.1. Quadro comparativo dos recursos de cada plataforma. ....</i>	<i>66</i>
<i>Tabela 4.2. Principais recomendações de usabilidade. ....</i>	<i>67</i>
<i>Tabela 4.3. Principais tags suportadas pelo framework Interop. ....</i>	<i>72</i>
<i>Tabela 5.1. Tipos de Mídia Existentes no FEB. ....</i>	<i>82</i>
<i>Tabela B.1 Conjunto de metadados propostos como extensão do padrão LOM, seção 4:Technical.....</i>	<i>104</i>
<i>Tabela C.1: Conjunto de metadados propostos como extensão do padrão LOM a partir de metadados do padrão TV-Anytime. ....</i>	<i>108</i>

## RESUMO

Até pouco tempo, o computador pessoal era o único dispositivo disponível para acesso a conteúdo digital. Com a introdução da TV digital interativa no Brasil e a evolução dos aparelhos celulares, essas plataformas se tornaram alternativas de acesso em momentos onde não está presente um computador e também como opção para a população de menor poder aquisitivo, visto que são dispositivos mais baratos.

Porém, o desenvolvimento de objetos de aprendizagem ainda continua sendo pensado para uma única plataforma, desperdiçando grande parte do potencial de uso. Quando raramente são previstos para mais de uma plataforma, o desenvolvimento de cada versão é realizado de forma isolada, gerando redundância de conteúdo e elevando desnecessariamente o custo de criação e manutenção.

Nesse contexto, este trabalho traz uma nova abordagem visando a criação de objetos de aprendizagem interoperáveis, ou seja, desenvolvidos de forma que o mesmo conteúdo possa ser executado nas plataformas Web, TV digital e dispositivos móveis. Para isso, inicialmente foram identificados os recursos e restrições existentes em cada uma das plataformas citadas, assim como as principais recomendações de usabilidade. O resultado desse estudo gerou as seguintes recomendações: a) mecanismo de construção de conteúdo uma única vez de forma que ele se adapte para todas as plataformas; b) mecanismos de adaptação da mesma mídia visando seguir critérios de usabilidade de cada plataforma (ex: tamanho e cor do texto); c) mecanismos de reconhecimento de cada plataforma e envio da mídia adequada para cada uma.

Outro aspecto complementar tratado foi em relação à catalogação de objetos de aprendizagem, uma vez que os padrões de metadados educacionais existentes não prevêm o uso de objetos de aprendizagem por diferentes plataformas. Em função dessa necessidade, realizou-se o estudo dos principais padrões de metadados educacionais, assim como os utilizados nas plataformas Web e de TV digital. Como resultado, duas extensões foram propostas aos padrões de metadados educacionais, possibilitando: a) indicar em quais plataformas é possível utilizar o objeto de aprendizagem e b) criar segmentos lógicos de um objeto de aprendizagem e, opcionalmente, agrupá-los por características em comum.

Para validação, foram efetuadas algumas implementações de diferentes objetos de aprendizagem. Esses objetos de aprendizagem também foram catalogados com as extensões de metadados propostas, exemplificando seu uso.

**Palavras-Chave:** TV digital, Interoperabilidade, Padrões de Metadados, Objetos de Aprendizagem.

# **A Proposal for Interoperable Learning Objects Construction and Cataloguing**

## **ABSTRACT**

Until recently, the personal computer was the unique device available for accessing digital content. With the introduction of interactive digital television in Brazil and the evolution of mobile phones, these platforms have become alternatives for content accessing in moments where the personal computer is not available. Additionally, it is an option for people with less purchasing capability, since they are cheaper devices.

However, development of learning objects is still being designed for a single platform, wasting much of its potential usage. When rarely provided for more than one platform, the development of each version is performed in isolation, creating redundant content and unnecessarily raising the cost of construction and maintenance.

In this context, this dissertation presents a new approach towards the creation of interoperable learning objects, i.e., developed in a way that the same content can be executed over the Web, digital television, and mobile devices. For that, the resources and restrictions for the above platforms were initially identified, as well the main interface usability recommendations. The result of this study generated the following recommendations: a) mechanisms to create the content just once in a way that adapts itself for each platform; b) mechanisms for media adaptation, following usability recommendations for each platform (font size and color, for example); c) mechanisms to recognize client platform and send the adequate media.

Another complementary aspect that was considered is learning object cataloguing, since the existing educational metadata standards do not foresee the usage of learning objects towards different platforms. Based in this need, the study of main educational metadata standards was done, like as those used in Web and digital television. As result, two extensions were proposed to the educational metadata standards, allowing: a) the indication of in which platform it is possible to use the learning object and b) the creation of learning object logical segments and, optionally, the possibility grouping themselves by common features.

For validation, some different learning objects implementations were performed. Those learning objects have also been cataloged with the proposed metadata extensions, illustrating their use.

**Keywords:** Digital Television, Interoperability, Metadata Standards, Learning Objects.

# 1 INTRODUÇÃO

O mundo, cada vez mais interconectado, proporciona à população o acesso a serviços digitais através de diversos meios, como Web, TV digital e aparelhos móveis. Com base nessa evolução, pode-se prover conteúdo educacional para cada uma dessas diferentes plataformas (computador na Web, TV digital ou dispositivos móveis), permitindo ao cidadão se aperfeiçoar onde se encontre e independentemente de qual tecnologia esteja utilizando.

Seguindo essa tendência, este trabalho busca alternativas para possibilitar o desenvolvimento de objetos de aprendizagem interoperáveis, onde se produz o conteúdo uma única vez e o mesmo pode ser executado nas três plataformas citadas, evitando redundâncias.

Conceituando os termos utilizados, um objeto de aprendizagem é qualquer entidade, digital ou não, que possa ser usada, reusada e referenciada durante algum processo de aprendizagem (IEEE, 2002). Essa conceituação também é adotada neste trabalho, porém, limitando-se às entidades digitais.

Já o termo interoperabilidade é utilizado nesta dissertação como a capacidade da utilização de uma linguagem de programação, padrão ou protocolo comum, independente do dispositivo, de forma que um conteúdo possa ser utilizado em diferentes ambientes.

Assim, o foco é no reaproveitamento e adaptação entre as diferentes plataformas ao invés da criação e manutenção de objetos de aprendizagem de forma exclusiva para cada uma. Como vantagens do presente estudo, destacam-se: a) facilidade na construção e manutenção de conteúdo, pois utiliza uma base única; b) eliminação de redundância; c) diminuição da necessidade de espaço de armazenamento.

A aplicação na área da educação foi escolhida por acreditar-se que é onde este trabalho pode contribuir muito. O professor, com o apoio técnico necessário, gera o conteúdo de um determinado curso uma única vez, já no padrão interoperável, e as futuras alterações no curso serão efetuadas também de forma única, minimizando esforços. Criando objetos de aprendizagem no formato interoperável, obtém-se acesso a uma gama muito grande de usuários, já que envolve diretamente as três plataformas. Assim, pessoas que não tem acesso a computadores pessoais, poderiam obter os objetos de aprendizagem por meio da televisão digital ou telefones móveis, terminais mais baratos em relação aos computadores pessoais.

Além disso, essa solução não apenas aumenta a abrangência em relação ao número de usuários, como também possibilita estar em contato com o material instrucional por mais tempo, acessando-os nos diversos locais. Por exemplo, um aluno que inicia acessando um curso pelo computador pessoal na escola, continua o mesmo durante a



volta para casa, no ônibus por meio do celular e, ao chegar em casa, utiliza a TV digital para prosseguir seu estudo.

Além da preocupação com a etapa de criação, também se faz necessário disponibilizar um conjunto de informações sobre os objetos de aprendizagem que permita catalogá-los, prevendo seu uso pelas diferentes plataformas. A catalogação geralmente se dá por padrões metadados, que definem informações a respeito do objeto de aprendizagem e facilitam sua busca, avaliação, aquisição e uso, tanto por aprendizes e instrutores como por processos automatizados de *software* (IEEE, 2002). Nesta dissertação, a proposta para catalogação de objetos de aprendizagem segue essa definição, limitando-se à definição dos metadados adicionais necessários para atender também aos requisitos de interoperabilidade entre as diferentes plataformas.

A dificuldade atual se deve ao fato que os padrões de metadados educacionais existentes foram propostos inicialmente para a Web, sem prever o acesso de um mesmo objeto de aprendizagem por diferentes plataformas ou em diferentes formatos. Além disso, os padrões de metadados utilizados para a Web não são os mesmos utilizados para a TV digital.

Dessa forma, este trabalho propõe também dois conjuntos de metadados de extensão aos padrões educacionais, visando atender aos requisitos técnicos de cada plataforma, além de possibilitar a catalogação de objetos de aprendizagem previstos para diferentes plataformas.

Cabe ressaltar desde já que tanto o estudo da criação de objetos de aprendizagem, quanto de catalogação dos mesmos, limitou-se à análise da viabilidade técnica. A avaliação educacional e pedagógica, assim como entrevistas sobre a satisfação de uso de tais objetos de aprendizagem nas diferentes plataformas não fizeram parte do escopo desta dissertação, ficando como trabalhos futuros.

Este trabalho encontra-se inserido no projeto OBAA (Objetos de Aprendizagem Baseados em Agentes) (OBAA, 2010), executado na UFRGS pelo CINTED (Centro Interdisciplinar de Novas Tecnologias na Educação) e pelo laboratório PRAV (Projetos em Áudio e Vídeo), do Instituto de Informática. Dentre os objetivos do projeto OBAA, há dois deles que esta dissertação está alinhada: (1) apontar soluções que permitam o acesso a objetos de aprendizagem a partir das plataformas Web, móvel e de TV digital, e (2) propor um padrão de metadados nacional para educação que contemple a convergência de terminais de acesso.

## 1.1 Objetivos

Este trabalho tem como objetivo geral investigar alternativas para criação e catalogação de objetos de aprendizagem de forma que possam ser executados nas plataformas Web, móvel e de TV digital.

São objetivos específicos desta dissertação:

- Estudar os principais formatos básicos de representação eletrônica de mídias (textos, imagens, sons e animações), linguagens de programação e protocolos existentes nas plataformas Web, móvel e de TV digital;
- Estudar os principais padrões de metadados educacionais, assim como os existentes para Web e para TV digital;

- Identificar recursos em comum nas plataformas citadas, que permitam criar objetos de aprendizagem interoperáveis, ou seja, que permitam ao usuário fruir do conteúdo educacional digital, tanto no ambiente Web, móvel e de TV digital;
- Verificar a necessidade ou não de extensão dos padrões de metadados educacionais, visando: (a) possibilitar a catalogação de objetos de aprendizagem previstos para diferentes plataformas e (b) atender a requisitos de metadados necessários para as plataformas Web e de TV digital;
- Recomendar extensões de metadados aos metadados educacionais, caso seja identificada tal necessidade no item anterior;
- Criar novos e/ou adaptar objetos de aprendizagem já existentes para possibilitar a sua execução nas plataformas Web, móvel e de TV digital, demonstrando a viabilidade técnica de criação de objetos de aprendizagem interoperáveis;
- Catalogar os objetos de aprendizagem criados com os metadados propostos para demonstrar a utilização e forma de uso dos mesmos, apresentando evidências de sua utilidade.

## 1.2 Trabalhos Relacionados

Na área de interoperabilidade, a preocupação de construção de uma plataforma comum para acesso à informação por meio de diversos equipamentos não é um objetivo recente. Dentre os trabalhos da W3C (*World Wide Web Consortium*), hoje a organização mais respeitada em termos de padronização para a Web, está a meta “Interoperabilidade da Web”, que busca compatibilidade no acesso à Web por todos os dispositivos (W3C, 2009-a). A iniciativa da Web Móvel do W3C tem como meta tornar o acesso à Web, a partir de qualquer tipo de equipamento, tão simples, fácil e conveniente quanto o acesso a partir de um computador pessoal. Dessa forma, telefones celulares, *smartphones*, assistentes pessoais digitais, sistemas de televisão interativos, sistemas de resposta por voz, quiosques e até mesmo alguns eletrodomésticos podem acessar a Internet (W3C, 2009-a).

Para atingir o objetivo de uma Web única, as especificações para os formatos e protocolos da Web precisam ser compatíveis entre si e permitir que (todos) os equipamentos e *softwares* usados para acessar a Web funcionem juntos. Dessa forma, o W3C cria e promove formatos e protocolos interoperáveis abertos (não-exclusivos) a fim de evitar a fragmentação de mercado que ocorreu no passado (W3C, 2009-a). A Figura 1 ilustra as tecnologias recomendadas para fazer da Web futura uma infraestrutura robusta, escalável e adaptativa para o mundo da informação.

O presente trabalho segue as recomendações de protocolos indicados pela W3C para Web e móveis, mas vai além, analisando também o ambiente de TV digital e propondo formatos interoperáveis de mídia.

O foco de diversos trabalhos de pesquisa atualmente está relacionado à interoperabilidade entre celulares e televisão digital. É o caso do trabalho de Paulson (2006), que faz uma análise das possibilidades de banda e tipos de padrões existentes para assistir TV nos dispositivos móveis. Cagenius (2006) propõe uma abordagem onde o usuário transfere a visualização da TV digital para o celular, podendo também interagir através do dispositivo móvel. O trabalho de Liang (2007) apresenta a padronização para cenários futuros envolvendo o padrão DVB-S, integrando TV digital,

dispositivos móveis e Internet banda larga. O artigo aborda algumas necessidades de rede, como o uso de QoS e a padronização dos canais de retorno via satélite.

Porém, os trabalhos citados verificam a possibilidade do usuário assistir televisão no dispositivo móvel, enquanto que a proposta deste trabalho vai além, propondo também uma padronização para a geração do conteúdo, buscando interoperabilidade e otimização nos recursos. Além disso, este trabalho desenvolveu um *framework* para criação de conteúdo interoperável, recurso não apresentado nos trabalhos citados.

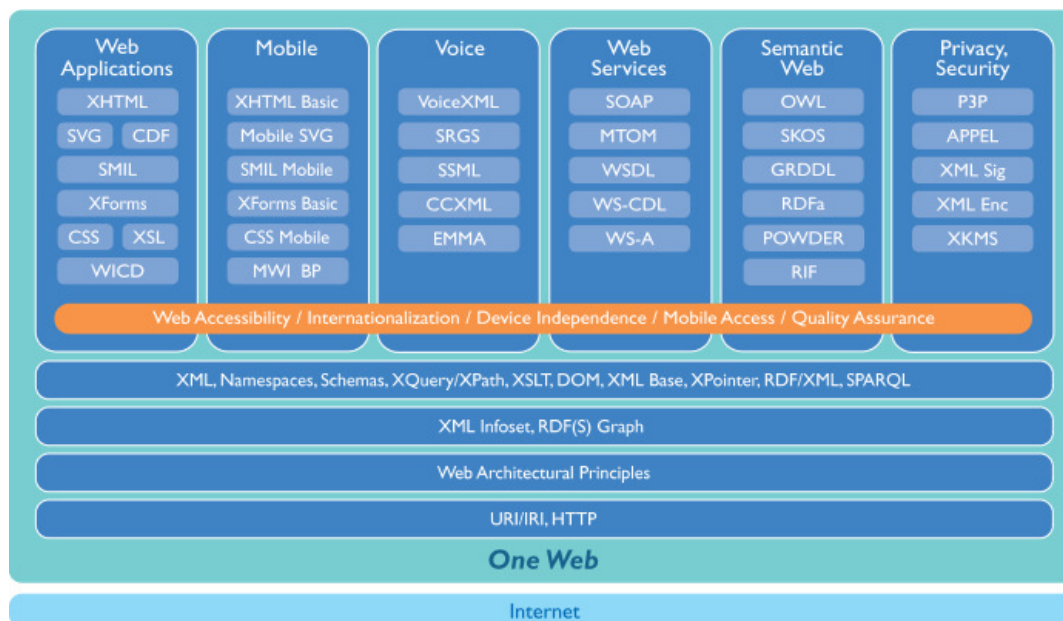


Figura 1. Recomendação de infra-estrutura interoperável para Web (W3C, 2009-a).

Em relação a trabalhos relacionados à especificação de um padrão unificado de metadados, a maioria dos projetos e sistemas que utilizam padrões complexos formados por vários sub-padrões, como o MPEG-21, utilizam apenas uma parte desses padrões. Até mesmo padrões mais simples como o *Learning Object Metadata* (que será detalhado mais adiante) possuem diversos perfis, onde cada perfil foi criado para atender a necessidades de um conjunto específico de aplicações. CanCore (2006) é um perfil LOM para atender aos requisitos canadenses, enquanto que o UK LOM Core é um perfil para o Reino Unido (JISC, 2010). Um ponto importante dos perfis é que os metadados que estão de acordo com um perfil também estão em conformidade com o padrão do qual o perfil foi originado, uma vez que são subconjuntos do mesmo.

Em outros casos um único padrão não atende as necessidades de uma aplicação, por isso são feitas extensões e integrações de padrões. Esta integração ainda é uma área de pesquisa em aberto, porém, algumas abordagens já foram propostas e implementadas.

Pfeiffer e Srinivasan (2000) integraram em uma arquitetura os padrões TV-Anytime e MPEG-7. Nessa arquitetura o padrão de metadados principal é o TV-Anytime, porém, quando há a necessidade de descrever algumas características de um conteúdo áudio-visual, tal descrição é realizada utilizando o padrão MPEG-7. Portanto Pfeiffer e Srinivasan definiram diretrizes e pontos onde se deve usar o TV-Anytime ou o MPEG-7. Como alguns dados do MPEG-7 também aparecem no TV-Anytime, a definição de diretrizes implicitamente definiu perfis dos dois padrões que são combinados de forma complementar. Dessa maneira são evitadas replicações de dados.

Já em relação a metadados de interoperabilidade, Durand, Kazai e Lalmas (2005), no projeto SAVANT (*Synchronised and Scalable AV content Across NeTworks*), tiveram como meta empregar simultaneamente *broadcast* e redes de telecomunicação de forma a possibilitar o uso da TV interativa através de aparelhos estáticos (televisão), portáteis (TabletPc) e móveis (PDA). Para isto foi explorada a integração do MPEG-7, MPEG-21 e TV-Anytime. O TV-Anytime foi utilizado como *framework* para, semanticamente, descrever serviços de TV digital, o MPEG-7 foi usado para descrever formatos de mídia e o MPEG-21 foi aplicado na definição de formas alternativas para exibição de conteúdo nas diferentes plataformas (aparelhos estáticos, portáteis e móveis).

Este trabalho reutiliza as ideias de harmonização de padrões de metadados, porém, avança ao aplicar os mesmos como extensão ao padrão educacional LOM, contemplando, em um conjunto unificado de metadados, tanto os requisitos educacionais como técnicos das plataformas Web, de TV digital e móvel (metadados de interoperabilidade e de segmentação).

Em relação ao projeto SAVANT, este trabalho reutiliza a solução de interoperabilidade proposta, porém expande a mesma para a plataforma Web e para outros dispositivos móveis (não contemplados no referido projeto) e adapta os metadados para aderir ao padrão LOM, que é a base dos padrões educacionais existentes.

### 1.3 Outras Soluções para Interoperabilidade

Além de trabalhos científicos, diversas soluções de mercados objetivam viabilizar o desenvolvimento unificado de *software* para diferentes plataformas.

A principal tentativa para atingir a interoperabilidade de aplicativos atualmente é o uso de máquinas virtuais, as quais procuram abstrair as diferenças de *hardware* e *software*, oferecendo um ambiente comum para execução de aplicativos, independente de plataforma. Uma máquina virtual pode ser definida como uma duplicata isolada de uma máquina real, normalmente é uma implementação em *software* que permite a execução de outros programas ou sistemas como se estivessem em um ambiente de execução diferente do que realmente estão. Portanto, para viabilizar isso, a mesma máquina virtual deve ter uma implementação para cada um dos ambientes de execução desejados.

A máquina virtual Java (*Java Virtual Machine – JVM*) é uma das mais conhecidas, podendo ser utilizada em servidores, computadores pessoais, sistemas embarcados e de tempo real (SUN MICROSYSTEMS, 2010-a). Foi desenvolvida pela *Sun Microsystems* e serve como base para desenvolvimento de programas utilizando a linguagem Java. Porém, devido às grandes diferenças existentes entre as plataformas de execução, foram definidas diferentes edições. A edição padrão é a *Java Platform, Standard Edition* (JavaSE) (SUN MICROSYSTEMS, 2010-a), destinada a aplicações *desktop* para computadores pessoais e aplicações Web chamadas de *Applets*. A *Java Platform, Enterprise Edition* (JavaEE) (SUN MICROSYSTEMS, 2010-b) é destinada a sistemas de arquitetura orientada a serviços e aplicações Web complexas. A *Java Platform, Micro Edition* (JavaME) (SUN MICROSYSTEMS, 2010-c) é destinada a dispositivos pequenos e/ou com poucos recursos de *hardware*, incluindo celulares, PDAs (*personal digital assistants*), *set-top boxes* e impressoras. Devido à variedade de dispositivos móveis, o Java ME subdivide-se ainda em diferentes perfis, os quais serão detalhados na

seção referente aos dispositivos móveis. O JavaFX (SUN MICROSYSTEMS, 2010-d) é uma proposta recente da SUN como evolução das edições Java para competir na criação e distribuição de *Rich Web Applications* (RIAs), prometendo também permitir a visualização desses aplicativos nas telas de qualquer dispositivo, independente de plataforma. Porém, embora seja uma promessa de interoperabilidade, o JavaFX, como qualquer edição Java, depende da máquina virtual embarcada nos diversos dispositivos e, portanto, dependerá da adoção pelos diversos fabricantes de equipamentos.

Com propósitos semelhantes aos do JavaFX, o *Open Screen Project* (ADOBE, 2009) é uma iniciativa conduzida pela Adobe em parceria com diversas empresas, que possui como visão permitir aos consumidores interagir com experiências de Internet rica naturalmente entre qualquer dispositivo, em qualquer lugar. Com esse objetivo, diversos parceiros estão trabalhando juntos para fornecer um ambiente de execução para navegação aberta e aplicações independentes. A esse ambiente foi dado o nome de Adobe AIR, baseado na linguagem Flash, cujo *player* está instalado em mais de 98% dos *desktops* conectados à internet (ADOBE, 2010-a).

Outra abordagem, iniciada com a popularização do acesso a Web pelo celular e a evolução de seus navegadores embarcados, é a adaptação do conteúdo e leiaute para as telas menores dos dispositivos móveis. Como exemplos de soluções disponibilizadas para esse tipo de adaptação são o Google<sup>1</sup> e o SKWEEZER<sup>2</sup>. O modo de uso é semelhante em ambas as soluções: o usuário acessa o site que realiza o serviço de conversão e digita o endereço Web original que deseja acessar, sendo redirecionado para uma versão adaptada da página requisitada. Embora os resultados ainda não sejam suficientemente satisfatórios e ainda voltados apenas para celulares, é uma técnica interessante e promissora, podendo futuramente ser estendida para os *browsers* outras plataformas, como a TV digital.

Em termos de soluções voltadas à educação, o projeto Amadeus (PROJETO AMADEUS, 2010) visa o desenvolvimento de um sistema de gestão de aprendizagem, permitindo estender as experiências adquiridas de usuário de educação a distância para diversas plataformas (Internet, *desktop*, celulares, PDAs e, futuramente, TV digital). Uma versão para computadores pessoais e outra para dispositivos móveis já está disponível como *software* livre no Portal do Software Público Brasileiro, podendo ser realizado o *download* mediante cadastro gratuito e ingresso na comunidade Amadeus (BRASIL, 2010). Porém, ainda não existe versão do sistema disponível para TV digital.

Percebe-se, portanto, que a maioria das abordagens existentes se baseia na criação de máquinas virtuais ou implementações específicas de ambientes para cada plataforma, abstraindo dos aplicativos as especificidades de *hardware* e sistema operacional. Já nesta dissertação, como um estudo base, estuda os recursos já disponíveis atualmente em cada plataforma, identificando tecnologias de uso em comum e que, se utilizadas, permitirão a interoperabilidade dos objetos de aprendizagem. Esse estudo se justifica, pois várias máquinas virtuais já foram propostas por grandes corporações, enquanto que não foi encontrado nenhum estudo semelhante ao deste trabalho, propondo o reaproveitamento das tecnologias já existentes e disponíveis.

---

<sup>1</sup> Disponível em: <<http://www.google.com/gwt/n>>. Acesso em: jul. 2010.

<sup>2</sup> Disponível em: <<http://company.skweezer.com/>>. Acesso em: jul. 2010.

## 1.4 Estrutura do Trabalho

Além deste capítulo de introdução, esta dissertação está organizada em seis capítulos e quatro apêndices.

O Capítulo 2 (Padrões Inerentes às Plataformas) apresenta as plataformas de computadores pessoais, TV digital e dispositivos móveis, descrevendo os principais recursos em termos de formatos de mídia e linguagens de programação. Maior ênfase foi dada a TV digital por ser a tecnologia mais recente e com maiores restrições em termos de tecnologias disponíveis. Apresenta ainda as principais recomendações de usabilidade de cada plataforma.

O Capítulo 3 (Padrões de Metadados) descreve os principais padrões de metadados utilizados atualmente para a educação, para Web e TV digital.

Com base na análise individual de cada plataforma e dos padrões de metadados estudados nos capítulos anteriores, o Capítulo 4 (Propostas para Interoperabilidade) apresenta: (a) recomendações para criação de conteúdo interoperável, composta por formatos de mídia e linguagens de uso comum, (b) o *framework Interop*, desenvolvido para facilitar a criação de objetos interoperáveis e (c) uma proposta de extensão dos padrões educacionais com dois novos grupos de metadados.

No Capítulo 5 (Implementações para Validar a Proposta) são apresentados três objetos de aprendizagem criados seguindo as recomendações para desenvolvimento interoperável, juntamente com seus metadados, objetivando validar a proposta de interoperabilidade nos diferentes ambientes disponíveis para testes.

O Capítulo 6 (Considerações Finais) apresenta os resultados obtidos com a experimentação descrita no Capítulo 5 e apresenta a análise desses resultados, elencando as principais contribuições e impactos do trabalho. Além disso, são apresentadas possíveis continuidades para o trabalho.

O Apêndice A (Terminais de TV Digital e Possíveis Cenários Educacionais) apresenta os principais recursos que poderão estar equipados os *set-top boxes* e como esses recursos podem ser utilizados no cenário educacional.

O Apêndice B (Metadados Técnicos Propostos) apresenta a especificação da extensão aos metadados técnicos propostos ao padrão LOM (IEEE, 2002), seguindo o mesmo formato da especificação do referido padrão.

O Apêndice C (Metadados de Segmentação Propostos) apresenta a especificação do conjunto de metadados de segmentação propostos ao padrão LOM (IEEE, 2002), seguindo o mesmo formato da especificação do referido padrão.

Apêndice D (Ferramentas Utilizadas) apresenta as ferramentas utilizadas para conversões de formatos de mídia e para montar o ambiente de validação dos objetos de aprendizagem desenvolvidos, juntamente com a justificativa da escolha de cada uma delas.

## 2 PADRÕES INERENTES ÀS PLATAFORMAS

Atualmente, os terminais que desfrutam de maior projeção, como meios de acesso ao conteúdo por parte dos usuários, são os computadores pessoais, telefones móveis e a TV digital (GRUPO TELEFÔNICA NO BRASIL, 2007). Baseado em padrões mundiais, este capítulo descreve o estudo sobre essas três plataformas. O objetivo é identificar os principais recursos e limitações de cada ambiente, assim como recomendações de usabilidade, que servirá como embasamento para identificação de recursos em comum, gerando recomendações para criação de objetos de aprendizagem interoperáveis, ou seja, que possam ser executados nas três plataformas citadas.

Visto que a plataforma de televisão digital brasileira é a mais recente, implantada pelo governo brasileiro em dezembro de 2007, e mais restritiva em termos de recursos e tecnologias disponíveis dentre as três, a mesma será enfatizada, apresentando um estudo mais detalhado para essa plataforma. Para apresentação dos recursos das plataformas de computadores pessoais e dispositivos móveis, maior ênfase será dada ao acesso de conteúdos digitais via navegadores Web, visto sua maior padronização em relação aos recursos disponíveis.

### 2.1 Computadores Pessoais

Com o advento da Internet, a Web vem se tornando a principal fonte para busca de conteúdos, devido a facilidade e agilidade de acesso, bastando ter um dispositivo com navegador Web e conexão com a Internet. Os computadores pessoais, conhecidos também como PCs (*personal computers*) ou *desktops*, e *notebooks* constituem a principal fonte de acesso a Web na atualidade.

Os navegadores Web, *web browsers* ou simplesmente *browsers*, são os aplicativos utilizados para o acesso a páginas da Web, permitindo ao usuário visualizar, interagir e navegar pelas páginas. Atualmente, os *browsers* mais utilizados são o Microsoft Internet Explorer (60,74%) e o Mozilla Firefox (22,91%) (MARKETSHARE, 2010).

Inicialmente os *web browsers* permitiam apenas a visualização de conteúdo texto, mas atualmente permitem a execução de diversos tipos de mídias, tais como imagens, vídeos, áudio, *scripts* e animações, onde formatos mais específicos geralmente requerem instalação de *plugins*<sup>3</sup>. Com a constante evolução que vem sendo apresentada pelas tecnologias utilizadas na fabricação desses equipamentos e o surgimento de *softwares* cada vez mais avançados torna-os sistemas com recursos abundantes, se

---

<sup>3</sup> *Plugin* (também conhecido por *plug-in*, *add-in*, *add-on*) é um programa de computador usado para permitir ao navegador Web realizar tarefas especiais ou específicas, como exibir formatos especiais de imagens ou tocar arquivos multimídia (MOZILLA, 2010).

comparados aos outros, visto que geralmente possuem vasta capacidade de armazenamento, processamento, dispositivos de interação e também amplo suporte a diferentes formatos de mídias. Por essas características, há cada vez menos limites para utilização de linguagens e recursos.

As próximas subseções apresentam os principais formatos de mídia utilizados nos computadores pessoais, com foco principal no acesso por meio de navegadores Web.

### 2.1.1 Áudio e Vídeo

Em relação ao suporte a áudio e vídeo na Web, o suporte é fornecido por meio de *plugins*, onde se pode citar como principais o *Windows Media Player* (MICROSOFT, 2010), *Real Player* (REAL NETWORKS, 2010), *Quick Time* (APPLE, 2010-b) e o *Adobe Flash* (ADOBE, 2010-a).

Dentre os principais formatos proprietários de áudio e vídeo, estão o WMV (*Windows Media Video*), WMA (*Windows Media Audio*) e ASF (*Active Streaming Format*) propostos pela Microsoft, sendo que o último para execução via *streaming*. Os formatos RM (*Real Media*) e RA (*Real Audio*) são proprietários da *Real Networks*. O *Quick Time* tem por formato padrão o MOV (abreviação de *Movie*), enquanto o *Flash* utiliza o formato FLV, que apenas encapsula outros formatos de áudio e vídeo.

Dentre os formatos abertos, os padrões mais comuns para vídeo são o MPEG (*Moving Picture Experts Group*) e seu sucessor MPEG-4, enquanto que para áudio, pode-se citar o MP3 (*MPEG Audio Layer 3*) e o seu sucessor AAC (*Advanced Audio Coding*).

Os *plugins* citados oferecem suporte a praticamente todos os formatos, mesmo aqueles proprietários propostos por empresas concorrentes.

### 2.1.2 Imagens

Não há limites nas especificações Web em relação ao formato gráfico que pode ser utilizado na Web. É necessário apenas um tipo MIME (*Multipurpose Internet Mail Extensions*) que o formato correto é identificado e um visualizador adequado (se existir ao menos um) pode ser localizado (W3C, 2003-b). Porém, na prática, alguns formatos são mais utilizados que os demais e são mais adequados para um determinado domínio. São formatos gráficos recomendados pelo W3C (W3C, 2003-b) o PNG (*Portable Network Graphics*), o JPEG (*Joint Photographic Experts Group*), o GIF (*Graphics Interchange Format*) e o SVG (*Scalable Vector Graphics*).

O formato GIF é preferencial para imagens que não necessitam grande número de cores, por apresentar tamanho reduzido sem ocorrência de perdas (COMPUSERVE, 1990).

O formato PNG (ROELOFS, 2010) é um formato de compressão de imagem sem perdas, livre de patentes, projetado para substituir o formato GIF e, como extensão, o muito mais complexo formato TIFF (*Tagged Image File Format*). Para a Web, o PNG tem três vantagens sobre o GIF: *alpha channels* (suporte a transparência variável), *gamma correction* (controle de brilho da imagem entre as diversas plataformas) e *interlacing* bi-dimensional (método de display progressivo). Ele também geralmente comprime melhor a imagem do que o formato GIF em quase todos os casos, mas a diferença fica apenas entre 5% e 25%. Em contrapartida o PNG não possui suporte a múltiplas imagens e animações.



O JPEG é um formato *lossy compression*, ou seja, de compressão que pode ocasionar perdas de qualidade (W3C, 2003-c). Porém, essas perdas geralmente são minimizadas por algoritmos específicos e o ganho em relação tamanho de arquivo é muito maior do que nos formatos sem perda (como o PNG), mesmo em níveis de compressão de alta qualidade da imagem (ROELOFS, 2010).

O SVG (W3C, 2010-b) é uma linguagem para descrever imagens bi-dimensionais e aplicações gráficas em XML (*Extensible Markup Language*). Sendo uma linguagem de imagem vetorial escalável, permite aumento de tamanho sem perdas de detalhes e animações. Assim, é uma tecnologia promissora, esperada como a nova geração de imagens para páginas Web. A versão atual do Firefox (versão 3.6.8) não possui suporte total a SVG, mas apresenta cobertura da maioria das funcionalidades (MOZILLA, 2009).

### 2.1.3 Extensible HiperText Markup Language (XHTML)

Embora os *web browsers* exibam conteúdo textual puro e também no formato XML, a linguagem declarativa básica para os *web browsers* é o HTML (*HyperText Markup Language*) e sua evolução, o XHTML (*Extensible HyperText Markup Language*).

O HTML (W3C, 1999) foi a primeira linguagem utilizada na criação de páginas Web, tendo sua origem juntamente com o surgimento da Web. Sua principal característica é o uso de *tags* para indicar elementos de estrutura visual nos documentos (leiaute), possuindo diversos tipos de elementos para esse fim, tais como imagens, tabelas, marcadores de parágrafos e criação de divisões hierárquicas.

Os principais componentes de um documento HTML são as *tags* (elementos) e os elementos textuais. Os elementos, por sua vez, possuem atributos e são divididos em três tipos: estruturais, de apresentação e hipertexto. Enquanto que os elementos textuais correspondem ao conteúdo texto que é exibido nas páginas.

Elementos do tipo estrutural subdividem os documentos, estruturando o conteúdo de acordo com seu motivo. Por exemplo, o trecho de código `<h2>Autores</h2>` demarca a palavra *Autores* como um cabeçalho de segundo nível. Outro exemplo de elemento estrutural é a *tag* `<div>`, que é muito utilizado para criar regiões nas páginas.

Outro tipo importante de elementos são os de apresentação, os quais permitem a customização e estilização do texto. As *tags* `<strong>` e `<em>` são exemplos desse tipo de elemento, formatando o texto em negrito e em itálico, respectivamente.

Já os elementos de hipertexto são utilizados para criar ligações (*links*) entre diferentes partes dos documentos, ou entre documentos diferentes. A Figura 2 exemplifica a criação de um *link* do documento atual para um documento cujo arquivo tem nome *documento.html*.

```
<a href="documento.html">Documento</a>
```

Figura 2. Exemplo de *tag* HTML para criação de *link* entre páginas. Adaptado de (W3C, 1999).

O XHTML (W3C, 2002) é a reformulação do HTML como uma aplicação XML, com correspondência aos elementos definidos no HTML 4.0. Portanto, é uma linguagem de marcação que tem a mesma abrangência de expressão do HTML, mas com uma semântica mais rígida, a qual deve seguir as leis da sintaxe XML (*Extensible*

*Markup Language*) (W3C, 2003-a). Como os documentos devem estar bem formatados, documentos XHTML corretos permitem seu processamento automático utilizando ferramentas XML padrão, ao contrário do HTML, o qual requer um *parser* específico, relativamente complexo e mais lento. Essa semântica fornece também a base para futuras extensões do XHTML, sem perder a compatibilidade com os agentes HTML já existentes (W3C, 2002).

Em 2001, o XHTML 1.1 se tornou uma recomendação W3C e, atualmente, é suportado pela maioria dos *browsers* existentes. Com isso, o XHTML vem se padronizando não apenas nos computadores pessoais, mas também nos *browsers* dos diversos dispositivos, como na televisão digital e dispositivos móveis.

No que se refere às *tags* suportadas, o XHTML 1.1 é equivalente ao HTML 4.0, e as diferenças entre as duas diz respeito à formatação do documento. A lista a seguir apresenta as principais recomendações do XHTML, onde algumas situações usuais no HTML não são mais permitidas no XHTML (W3C, 2002):

- Os documentos precisam ser bem formatados, seguindo a sintaxe o XML, ou seja, elementos aninhados devem ter seu escopo fechado dentro do elemento onde ele foi aberto:

*Exemplo de código errado:* `<p><em>texto enfatizado</p></em>`

*Código correto:* `<p><em>texto enfatizado</em></p>;`

- Todos os nomes de *tags* e de atributos deve ser escrito em letras minúsculas;
- Não pode haver *tags* abertas, ou seja, que não possuam sua *tag* correspondente de fechamento

*Exemplo de código errado:* `<p>parágrafo 1<p>parágrafo 2`

*Código correto:* `<p>parágrafo 1</p><p>parágrafo 2</p>`

- Todos os atributos devem ser circundados por aspas duplas

*Exemplo:* `<a href="proximaPagina.xhtml">clique aqui</a>`

- Não pode ser usada a minimização de atributos que era típica do HTML

*Exemplo de código errado:* `<dl compact>`

*Código correto:* `<dl compact="compact">`

- Elementos vazios também devem ser fechados.

*Exemplo:* `<br/>, <hr/>`

#### 2.1.4 Cascading Style Sheet (CSS)

O CSS (W3C, 2009-b) foi projetado para possibilitar a separação do conteúdo do documento (escrito em HTML ou uma linguagem de marcação similar como XHTML) da formatação da apresentação (escrito em CSS), com objetivo de melhorar a acessibilidade ao conteúdo, fornecendo maior flexibilidade e controle da especificação das características de apresentação, reduzindo a complexidade e repetição do conteúdo. O arquivo CSS é interpretado localmente pelo *web browser*, utilizando-o para definir cores, fontes, leiaute e outros aspectos da apresentação de documentos HTML e XHTML.

Dessa forma, o estilo e as informações de layout de um documento são definidos com as regras de estilos CSS, e são colocadas em separado do conteúdo do documento (em um arquivo externo). Para alterar a aparência de um documento, basta modificar a folha de estilos CSS que o mesmo utiliza.

As especificações CSS são mantidas pelo World Wide Web Consortium (W3C, 2009-b) e os navegadores atuais possuem suporte a esse padrão.

### 2.1.5 Linguagem *Script*

Tanto XHTML como CSS representam conteúdos estáticos, limitando a possibilidade de validação dos dados de usuário, criação de lógica dinâmica, matemática e procedural no lado do cliente. Para superar essas limitações das linguagens de marcação, os *browsers* embutem suporte linguagens de conteúdo imperativo, conhecidas como linguagem *script*.

O padrão existente hoje é o ECMA-262 (ECMAScript Release 3), criado para harmonizar linguagens *script* existentes, como os conhecidos JavaScript (Netscape) e JScript (Microsoft). O desenvolvimento da norma para ECMA-262 começou em novembro de 1996, sendo que a primeira edição dessa norma foi adotada pela Assembléia Geral da ECMA de junho de 1997 (ECMA, 2009).

A Web adota o padrão por completo. Outras plataformas adotam outros perfis que são subconjuntos desse padrão, conforme será descrito posteriormente na seção referente a cada plataforma.

### 2.1.6 Java

Java é uma das principais linguagens imperativas que podem ser utilizadas junto aos *web browsers*, sob a forma de *plugin*. A programação Java para Web nos computadores pessoais é feita pela criação de *Applets*. Um *Applet* é um programa escrito em linguagem de programação Java e que pode ser incluído em uma página HTML ou XHTML, de forma semelhante à forma que uma imagem é incluída em uma página (SUN MICROSYSTEMS, 2010-g). Para incluir um *Applet*, utiliza-se a tag `<APPLET>` e é necessário ter um *plugin* da máquina virtual Java instalada no *web browser*.

A Figura 3 ilustra um código exemplo de inclusão de um *Applet* em uma página Web (a) e a estrutura de código de um *Applet* (b), que apenas herda a classe `java.applet.Applet` e define o ciclo de vida do programa, por meio de quatro métodos representando eventos: *init* (na inicialização dos recursos necessários), *start* (quando inicia sua execução), *stop* (quando para a execução) e *destroy* (na finalização, para liberar recursos utilizados).

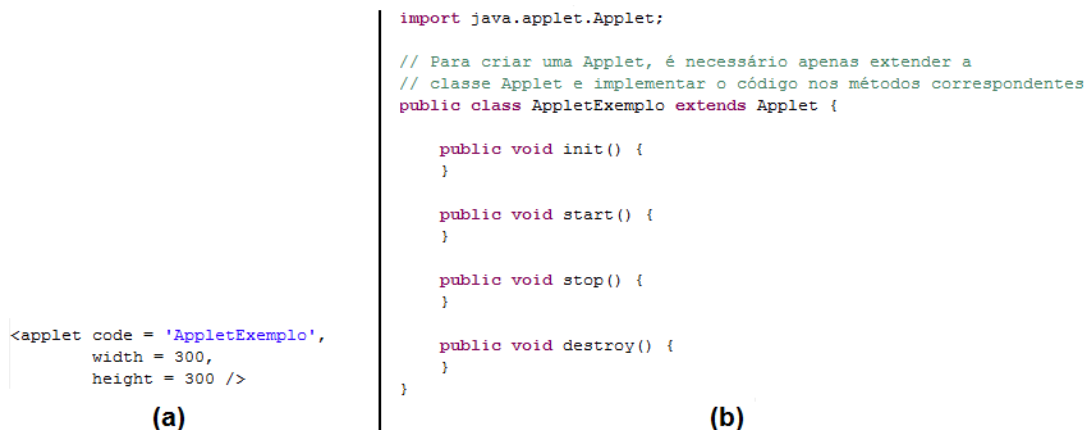


Figura 3. Exemplo de Applet em uma página Web (a) e estrutura do código Java do Applet (b). Adaptado de (SUN MICROSYSTEMS, 2010-g).

### 2.1.7 Flash

O Adobe Flash Player é uma máquina virtual instalada como *plugin* nos *web browsers* para executar arquivos com a extensão SWF. Conforme descrito anteriormente, o Flash Player é utilizado para *streaming* de áudio e vídeo, mas não apenas isso, ele possui uma linguagem *script* chamada *ActionScript* e permite de forma simples criar aplicativos para a Web, sendo por isso muito utilizado na área educacional para desenvolvimento de objetos de aprendizagem. Flash é também um formato comum para jogos, animações e interfaces embutidas em páginas Web.

Embora recentemente o SWF tenha se tornado um formato aberto novamente, a Adobe não dá sinais de disponibilizar o código fonte completo para o desenvolvimento como *software open source*. A principal alternativa gratuita para SWF é o *player* Gnash<sup>4</sup>, que ainda é bem incompleto no presente momento. Porém, como o SWF é agora um formato aberto, ele poderá ter uma qualidade bem maior com desenvolvedores implementando a especificação SWF oficial, sendo que anteriormente o desenvolvimento era realizado por engenharia reversa.

Além do *player*, existem algumas IDEs *open sources* para criação de animações e aplicativos de *eLearning* Flash, como o Salsaga<sup>5</sup> e o Open Dialect<sup>6</sup>.

Embora o Flash seja bastante utilizado na Web para desenvolvimento de objetos de aprendizagem (e por isso considerado neste capítulo), o mesmo não é previsto para TV digital no Brasil e, por esse motivo, não será considerado no presente momento como alternativa para desenvolvimento interoperável para as três plataformas.

### 2.1.8 HTML5

O HTML5 (W3C, 2010-a) é a próxima grande versão do padrão HTML. Assim como os antecessores (HTML 4.01 e XHTML 1.1), o HTML5 é um padrão para estruturar e apresentar conteúdo na Web.

<sup>4</sup> Disponível em: <<http://www.gnu.org/software/gnash/>>. Acesso em: jul. 2010.

<sup>5</sup> Disponível em: <<http://www.salasaga.org/>>. Acesso em: jul. 2010.

<sup>6</sup> Disponível em: <<http://dialect.openmodeling.net/wiki/>>. Acesso em: jul. 2010.

Em particular, o HTML5 inclui várias novas funcionalidades, incluindo elementos <video>, <audio> e <canvas>, projetadas para facilitar a inclusão de conteúdo gráfico e multimídia na Web, sem a necessidade de *plugins* específicos. Outros novos elementos como <section>, <article>, <header> and <nav> foram previstos para enriquecer a semântica dos documentos. Além disso, também estão previstas APIs para criação de aplicações offline, edição de documentos, geolocalização, banco de dados SQL, entre outros.

No momento, o HTML5 ainda se encontra em padronização e a versão atual dos navegadores Web implementam alguns dos novos recursos do HTML5. Da mesma forma, apenas os dispositivos móveis mais modernos já estão prevendo suporte a algumas funcionalidades do HTML5, enquanto que a TV digital não há padronização nesse sentido. Dessa forma, por ainda estar bastante incipiente, o HTML5 foi deixado como uma futura continuidade deste trabalho.

### 2.1.9 Usabilidade na Web

Na Web, em relação às fontes, sugere-se utilizar tipos padrões, presente na maioria dos navegadores e sistemas operacionais, preferencialmente: *Arial*, *Times New Roman*, *Verdana*, *Georgia* (THE INTERNET DIGEST, 2003).

Em relação às cores, são sugeridos textos pretos em fundos brancos sempre que possível ou, no caso de planos de fundo, utilizar fundos planos ou com padrões extremamente sutis (FORAKER DESIGN, 2010).

Outras recomendações gerais são apresentadas em (CHALMERS, 2008):

- Não utilizar mais que 3 (três) fontes na mesma página;
- Utilizar negrito para enfatizar trechos de texto, cor de fundo diferente também é aceitável (mas menos preferível);
- Evitar excessos de textos enfatizados e tamanhos de fonte muito discrepantes ao longo da mesma página;
- Sempre que a cor do texto não for preta então a cor do plano de fundo também deve ser especificada;
- Evitar cores quentes e/ou que tenham relação com sentimentos fortes (vermelho, por exemplo);
- Utilizar sublinhado e as cores padrões para *links*: azul para não visitados, roxo para visitados e vermelho para *links* com o foco do mouse;
- Evitar utilizar essas cores em outros textos do site.

## 2.2 TV Digital

A televisão digital terrestre e aberta no Brasil, denominada SBTVD (Sistema Brasileiro de Televisão Digital), teve suas primeiras transmissões oficiais em dezembro de 2007 e utiliza as especificações do ISDB-Tb (*Integrated Services Digital Broadcasting - Terrestrial – Brazilian Version*), um sistema novo, que foi desenvolvido em conjunto por diversos setores brasileiros como uma melhoria ao sistema-base, o ISDB-T (*ISDB Terrestrial Standard*).

Mesmo com a TV analógica, a maioria da programação veiculada pelas emissoras de televisão aberta já era produzida utilizando equipamentos de filmagem e armazenamento digitais (TAVARES, 2001). No entanto, como os segmentos de transmissão e recepção continuavam sendo analógicos, esse ganho de qualidade não chegava até os telespectadores. A introdução da tecnologia digital no serviço de televisão referiu-se, inicialmente, na digitalização desses dois segmentos, o que possibilitou de imediato, melhor qualidade de som e imagem para os telespectadores. Além dessa melhoria de qualidade, outra vantagem é a recepção móvel de sinais de televisão em carros, ônibus, trens e em telefones celulares das próximas gerações.

A transmissão de vários programas em um único canal é outra novidade que, em breve, possibilitará uma maior diversidade de conteúdos, atendendo às diferentes necessidades e interesses dos usuários. Na multiprogramação, os vários programas podem ser completamente independentes, levando a reaver o conceito de canal. Na TV analógica, o canal de 6 MHz (ou seja, o canal de frequência era confundido com o canal de programação (normalmente associado a uma rádio difusora). Com a TV digital, isso não é obrigatório, sendo que em um canal de frequência poderão ser transmitidos vários programas de TV diferentes (SOARES, 2009).

No entanto, a mudança mais esperada é a transformação do televisor em um equipamento interativo. Na televisão tradicional, os únicos tipos de interação possíveis são: trocar de canal, mudar o volume, ligar e desligar. Com a TV interativa o telespectador pode interagir para mudar não só o sinal de TV que está recebendo, mas poderá também executar diversas aplicações que representarão serviços de comércio (chamados de *tcommerce*), jogos, serviços governamentais (como voto eletrônico e declaração do imposto de renda), entre outros. Poderá, ainda, interagir escolhendo uma câmera em um jogo de futebol, participando de jogos de auditório, escolhendo suas preferências em aplicativos interativos como previsão de tempo, bolsas de valores, notícias de última hora e assim por diante.

Isso é viabilizado porque a TV digital possibilita o envio de dados (arquivos, programas de computador, etc.), além do áudio e vídeo, permitindo também a execução e a interação com aplicativos por meio do controle remoto da televisão. Dessa forma, a televisão interativa é a fusão da televisão tradicional passiva com tecnologias de computação, de forma a permitir que o telespectador interfira no que está vendo.

### **2.2.1 Interatividade na TV Digital**

A capacidade de interatividade da TV digital se deve à presença de três elementos principais: o multiplexador, o carrossel de dados e o terminal de acesso interativo (FERNANDES; LEMOS; ELIAS, 2004).

O multiplexador unifica os fluxos de áudio, vídeo e dados em um fluxo único de dados, que viabiliza sua transmissão em *broadcast* da emissora até o aparelho receptor do usuário. Esse fluxo compõe os eventos, programas de TV e aplicativos, os quais por sua vez compõem os serviços consumidos pela audiência. Cabe aqui explicitar a distinção utilizada entre os termos programa e aplicativo. Assim como utilizado por Soares (2009), será adotado aqui a seguinte convenção: a programação televisiva será chamada simplesmente de programa ou programa de TV, enquanto que um programa computacional será chamado de aplicação ou aplicativo. Observa-se, portanto, que o multiplexador poderá unificar o áudio e vídeo de um programa de TV com aplicativos interativos, que por sua vez serão transmitidos aos usuários.

O carrossel de dados é o responsável por transformar uma estrutura de arquivos e diretórios em um fluxo elementar, no formato em que o multiplexador possa capturar esses dados e transmitir juntamente com o áudio e vídeo (FERNANDES; LEMOS; ELIAS, 2004). Na televisão, a transmissão é unidirecional, em *broadcast*, o que torna impossível enviar os aplicativos de uma única vez, visto que cada usuário pode ligar a televisão ou escolher determinado canal em um momento diferente. Assim, para solucionar esse problema, foi criado o padrão DSM-CC (*Digital Storage Media, Command and Control*), que realiza a transmissão da estrutura de arquivos através do carrossel de dados, conforme ilustrado na Figura 4. Nesse padrão, um canal fica transmitindo dados em *broadcast* constantemente, de forma circular. Os arquivos são transmitidos um após o outro, até que todos os arquivos sejam enviados, quando então o envio recomeça do primeiro. Por esse motivo, o processo é conhecido como carrossel de dados.

A vantagem do carrossel de dados está no fato de não ser necessária uma conexão. Os dados são enviados em *broadcast* e, portanto, não necessitam de um servidor dimensionado para suportar múltiplas conexões simultâneas para esse fim.

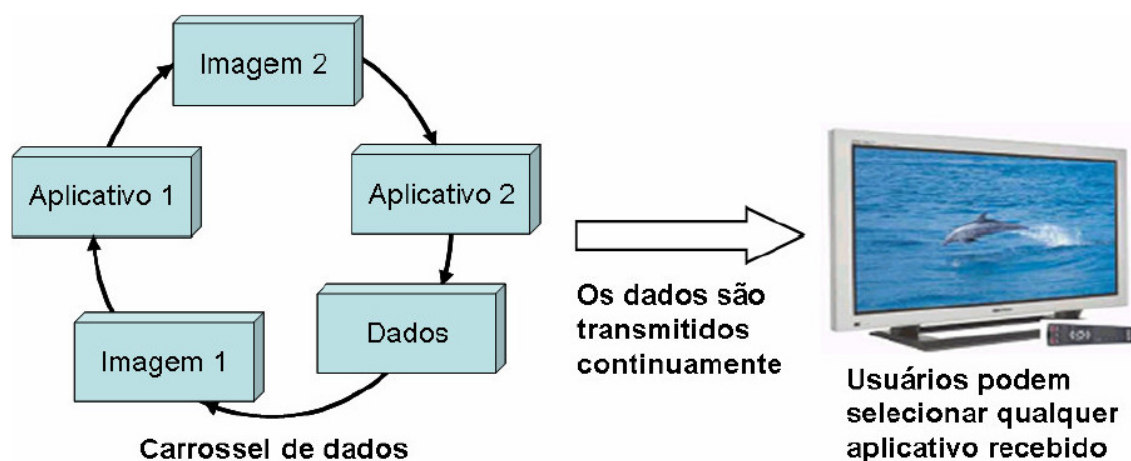


Figura 4. Carrossel de dados na TV digital. Adaptado de (BECKER et al, 2006).

O terminal de acesso é o responsável por receber o áudio, vídeo e outros dados da emissora e apresentar ao usuário. Ele possui capacidade de processamento, sendo capaz de interpretar computacionalmente os fluxos de dados multiplexados. Desse modo, o terminal de acesso executa uma aplicação que, por sua vez, exibe na TV uma interface com o usuário. Isto permite à audiência interagir com o programa de televisão através do controle remoto. Ao entregar à audiência um fluxo de dados localmente computável, a TV digital se torna interativa (FERNANDES; LEMOS; ELIAS, 2004). Esse sistema é ilustrado pela Figura 5, no qual a emissora gera o áudio, vídeo e a aplicação, converte para um fluxo e realiza a transmissão (nesse caso, transmissão terrestre). O terminal de acesso recebe o sinal digital por meio de uma antena e converte os dados recebidos para o formato analógico, que é apresentado na televisão convencional.

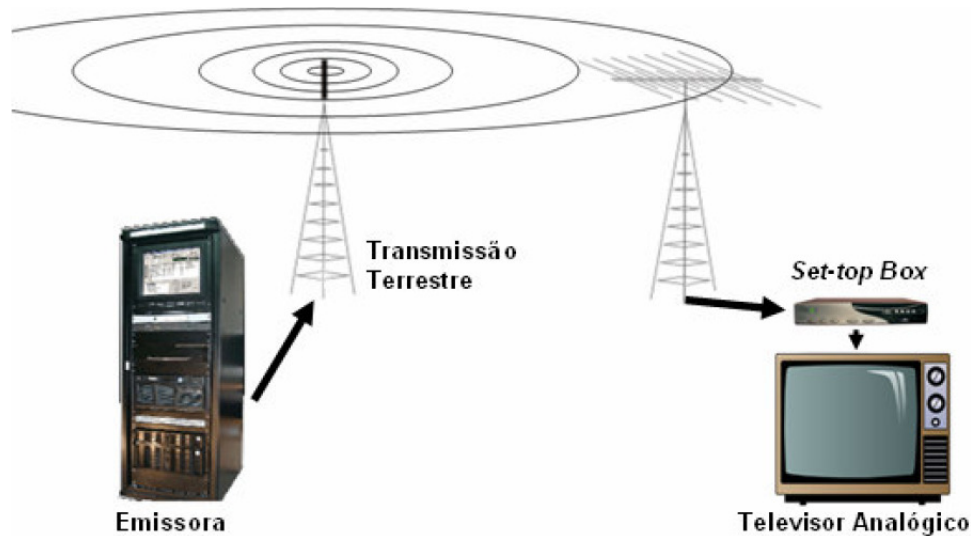


Figura 5. Transmissão e recepção de sinal de TV digital terrestre. Adaptado de (LEITE, 2005).

Dessa forma, o terminal de acesso é o dispositivo físico que engloba as funcionalidades necessárias, no lado do usuário, para a recepção do sinal e acesso aos serviços de uma plataforma de televisão digital.

Embora possa existir uma variedade de dispositivos de acesso à TV digital, o SBTVD classifica dois perfis: o *one-seg* e o *full-seg*. O receptor do tipo *full-seg* é o dispositivo capaz de decodificar informações de áudio, vídeo e dados contidas na camada do fluxo de transporte de 13 segmentos destinada ao serviço fixo (*indoor*) e móvel. Essa classificação é aplicada aos conversores digitais, também conhecido por *set-top box*, e aos receptores de 13 segmentos integrados com tela de exibição, mas não exclusivos a esses (ABNT, 2008-f). O receptor *one-seg* é o dispositivo que decodifica exclusivamente informações de áudio, vídeo, dados, etc., contidas na camada “A” locada no segmento central dos 13 segmentos. Essa classificação é destinada aos receptores do tipo portátil, também conhecido por “*handheld*”, especialmente recomendados para telas de exibição de dimensões reduzidas, normalmente até 7 polegadas. Entre os produtos classificados como *one-seg*, estão os receptores integrados com telefone celular, PDA, *dongle* (dispositivo no formato de um *pendrive* que é conectado na porta USB e permite ver televisão digital no computador) e televisores portáteis, os quais são energizados por uma bateria interna e, portanto, sem necessariamente demandar uma fonte externa de energia, bem como aqueles destinados a veículos automóveis. Este tipo de receptor é capaz de receber e decodificar apenas sinais de televisão digital terrestre transportado na camada “A” do fluxo de transporte, e, conseqüentemente, apenas sinais de perfil básico destinado aos dispositivos portáteis de recepção (ABNT, 2008-f).

O conhecimento dessa classificação é importante, pois alguns dos requisitos para cada um desses perfis são diferenciados, ou seja, alguns dos requisitos obrigatórios em um receptor *full-seg* são opcionais ou inexistentes no receptor *one-seg*.

Embora exista essa separação, considerou-se útil categorizar diferentes configurações de *set-top boxes* em função de recursos disponíveis ou não nos mesmos, levando a uma melhor compreensão dos possíveis cenários de uso da TV digital na área educacional. Essa classificação é apresentada no Apêndice A, onde as diferentes configurações são identificadas, além dos cenários de uso como um meio educativo.



### 2.2.2 O *Middleware* do SBTVD e Ambientes Declarativo e Procedural

Para tornar as aplicações de TV digital independentes da plataforma de *hardware* e *software* de cada fabricante de receptor e dar melhor suporte às aplicações voltadas para a TV, uma nova camada é acrescentada aos padrões de referência de um sistema de TV digital, denominada *middleware* (SOARES, 2009). Dessa forma, o *middleware* Ginga foi padronizado para a televisão digital brasileira, onde dois ambientes de programação estarão disponíveis (ABNT, 2008-g): o declarativo, baseado na linguagem NCL (Ginga-NCL), e o procedural, que utiliza uma máquina virtual Java (Ginga-J). A arquitetura do *middleware* Ginga é ilustrada na Figura 6.

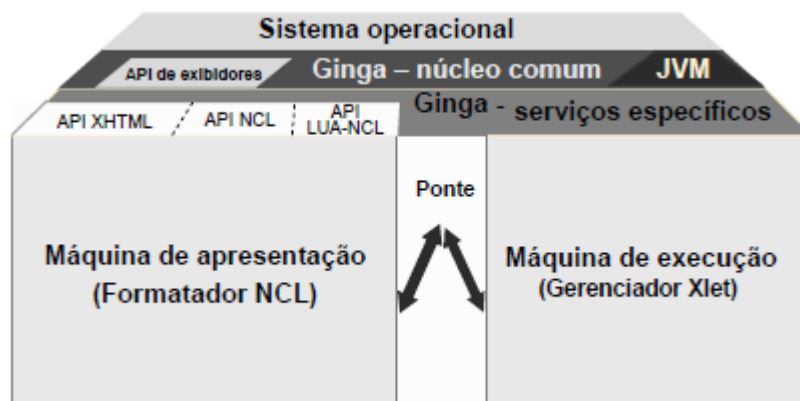


Figura 6. Arquitetura do *middleware* Ginga (ABNT, 2008-b).

No centro do *middleware* está o *Ginga Common Core* (Núcleo Comum Ginga), que implementa as funcionalidades centrais do *middleware* (como decodificadores de conteúdo comum, obtenção dos conteúdos transportados em fluxo de transporte e o suporte ao modelo conceitual de exibição), funcionalidades essas que servirão tanto para as aplicações procedurais quanto declarativas, sempre por meio da linguagem Java ou NCL, respectivamente (ABNT, 2008-b).

A máquina de apresentação tem como base o subsistema lógico Ginga-NCL, que é responsável pelo processamento de documentos NCL. A máquina de execução é o subsistema Ginga-J, responsável pelo processamento de conteúdos ativos, baseado na linguagem Java. Esses subsistemas serão tratados em mais detalhes nas subseções 2.2.5, 2.2.6 e 2.2.7.

### 2.2.3 Áudio e Vídeo

O SBTVD adotou o padrão MPEG-4 AAC para codificação do áudio principal de um programa. A codificação de áudio deve obrigatoriamente ser compatível com a ISO/IEC 14496-3. Os seguintes perfis e níveis do padrão MPEG-4 AAC devem obrigatoriamente ser permitidos para receptores *full-seg* (ABNT, 2008-b):

- LC (*low complexity*), perfil básico do padrão AAC, níveis L2 e L4;
- HE (*high efficiency*), perfil avançado de alta eficiência, combinando o perfil LC com o uso da ferramenta SBR (*spectral band replication*) para a versão 1 deste perfil, níveis L2 e L4;
- HE combinado à ferramenta PS (*parametric stereo*) para a versão 2 deste perfil; nível L2.

- Os receptores *one-seg* deve obrigatoriamente adotar a versão 2 do MPEG-4 AAC-HE (*high efficiency*).

Além do formato MPEG-AAC, é ainda opcional a implementação dos formatos PCM (AIFF-C) e MP3 (ABNT, 2008-f).

Para o vídeo, o sistema brasileiro de TV digital utiliza a codificação H.264, também conhecida como MPEG-4 AVC (*Advanced Video Coding*). Esse padrão de codificação é sucessor do MPEG-2, utilizado pelos demais padrões mundiais de TV digital.

“O H.264 contém várias facilidades novas, que permitem uma compressão de vídeo muito mais eficiente e flexível. As técnicas empregadas fazem do H.264 um padrão significativamente melhor do que os padrões anteriores, sob uma variedade de circunstâncias e ambientes de aplicação, em particular o ambiente de TV digital, onde ele oferece um desempenho bem melhor do que o MPEG-2 vídeo. Em especial, nas situações de alta resolução e alta taxa de *bits*, o padrão H.264, para a mesma qualidade de vídeo, gera uma taxa 50% ou ainda menor do que a taxa gerada pelo padrão MPEG-2.” (SOARES, 2009).

A Tabela 2.1 resume as principais configurações de vídeo do SBTVD.

Tabela 2.1: Codificação de vídeo no sistema brasileiro de TV digital.

	Receptores Fixos e Móveis	Receptores Portáteis
Padrão	ITU-T H.264 (MPEG-4 AVC)	ITU-T H.264 (MPEG-4 AVC)
Número de linhas	480 (4:3 e 16:9), 720 (16:9), 1.080 (16:9)	SQVGA (160 x 120 ou 160 x 90), QVGA (320 x 240 ou 320 x 180) e CIF (352 x 288); todos em 4:3 e 16:9
Taxa de Quadros	30 e 60 Hz	15 e 30 Hz

Fonte: SOARES, 2009.

Para transmitir o áudio e vídeo multiplexados em um carrossel de objetos, cada arquivo deverá ser codificado em formato “TS” (ISO/IEC 13818 apud ABNT, 2008-d).

#### 2.2.4 Imagens

Quanto às imagens para a TV digital, elas dependem do *middleware* Ginga para seu suporte. Todos os *set-top boxes* que tenham o Ginga embarcado deverão obrigatoriamente dar suporte aos formatos PNG, JPEG e H.264/MPEG-4 AVC “I-Picture”. O formato MNG é obrigatório em dispositivos *full-seg* (*set-top boxes* e televisores integrados) e opcional nos dispositivos *one-seg* (*handhelds*). Já os formatos GIF, MPEG-2 “I-Frame” e MPEG-4 “I-VOP” são opcionais (ABNT, 2008-b).

#### 2.2.5 Ambiente Declarativo

Embora a maioria dos ambientes declarativos dos principais sistemas de TV digital terrestre seja baseada em XHTML (W3C, 2002) e ECMAScript (ECMA, 1999), como nos *middlewares* ACAP<sup>7</sup> (linguagem ACAP-X), MHP<sup>8</sup> (DVB-HTML) e ARIB<sup>9</sup> (BML), o SBTVD escolheu o NCL (*Nested Context Language*) como linguagem declarativa.

<sup>7</sup> Disponível em: <<http://www.atsc.org/>>. Acesso em: ago. 2010.

<sup>8</sup> Disponível em: <<http://www.mhp.org/>>. Acesso em: ago. 2010.

A NCL tem uma separação mais acurada entre o conteúdo e a estrutura, não definindo nenhum objeto de mídia em si. Em vez disso, ela define a cola que prende os objetos de mídia em apresentações multimídia (SOARES, 2009). Como linguagem de cola, ela não restringe ou prescreve os tipos de conteúdo dos objetos de mídia. Assim, pode-se ter objetos de imagem, vídeo, áudio, texto, outros códigos declarativos ou imperativos sendo referenciados pelo código NCL. O suporte aos diferentes formatos de mídia, porém, dependerá de exibidores de mídia acoplados ao formatador NCL (exibidor NCL, renderizador de documentos, agente do usuário e *player* são alguns dos nomes atribuídos ao formatador de documentos) (ABNT, 2008-g) (SOARES, 2009).

Embora a linguagem declarativa seja a NCL, como linguagem de cola ela suporta mídias no formato XHTML, sendo que a implementação de um *player* para esse formato é obrigatória, tendo também como objetivo garantir a interoperabilidade com os demais sistemas (ABNT, 2008-g).

### 2.2.6 Linguagem Script

Em relação à linguagem de *script*, o SBTVD escolheu a linguagem Lua, com desempenho muito superior à ECMAScript em todos os quesitos importantes para uma aplicação em TV digital (SOARES, 2009), mantendo o ECMAScript como opcional (ABNT, 2008-g).

Lua é uma linguagem de programação funcional e imperativa, procedural, pequena e leve, projetada para expandir aplicações em geral, para ser usada como linguagem extensível e para ser embarcada em *softwares* complexos (ABNT, 2008-g). Lua combina programação procedural com poderosas construções para descrição de dados, baseadas em tabelas associativas e semântica extensível. É tipada dinamicamente, interpretada a partir de *bytecodes*, e tem gerenciamento automático de memória com coleta de lixo. Essas características fazem de Lua uma linguagem ideal para configuração, automação (*scripting*) e prototipagem rápida, característica de grande importância para as aplicações de TV, cujo desenvolvimento rápido é fundamental. Com seu núcleo pequeno e totalmente ANSI C, ela é facilmente embarcável, onde seu interpretador é uma biblioteca C, embarcável em várias linguagens, entre elas C/C++ (linguagem da implementação de referência do Ginga-NCL) e Java (linguagem do ambiente imperativo Ginga). Lua é hoje a linguagem de *script* padrão no seguimento de entretenimento. Alguns dos jogos mais famosos da atualidade utilizam Lua (SOARES, 2009).

Além de objetos NCLua (com código Lua), a NCL permite objetos com outros códigos imperativos como, por exemplo, objetos NCLet com código Java (Xlets), parte da ponte entre o ambiente declarativo e o ambiente imperativo do *middleware* Ginga.

Além disso, uma aplicação declarativa pode fazer referência a um código do subsistema procedural Ginga-J. Da mesma forma, uma aplicação procedural pode fazer referência a uma aplicação declarativa, contendo, por exemplo, conteúdo gráfico, ou pode construir e iniciar a apresentação de aplicações com conteúdo declarativo. Portanto, ambos os tipos de aplicação Ginga podem utilizar as facilidades dos ambientes de aplicação declarativo e procedural (ABNT, 2008-g).

---

<sup>9</sup> Disponível em: <<http://www.arib.or.jp/english/>>. Acesso em: ago. 2010.

### 2.2.7 Java

O Ginga-J é o subsistema lógico com linguagem imperativa do sistema Ginga, baseado na linguagem Java. Um componente-chave do ambiente de aplicação procedural é a máquina de execução do conteúdo procedural, composta por uma máquina virtual Java (ABNT, 2008-g).

Embora inicialmente a escolha natural fosse pelo GEM (DVB, 2008), uma solução de *middleware* já adotada por diversos padrões de TV digital do mundo, o Fórum SBTVD, numa parceria com a SUN Microsystems, optou por criar uma nova especificação, o JavaDTV. Essa substitui bibliotecas que poderiam incorrer em royalties futuramente, como o HAVi Level 2 UI para construção da interface (HAVI, 2009) e o DAVIC para gerenciamento de recursos escassos (DAVIC, 2009), por equivalentes funcionais, além de especificar funcionalidades adicionais aos outros padrões, como o suporte a multidispositivos (SOARES, 2009).

Os dispositivos *one-seg* devem obrigatoriamente implementar o subsistema declarativo, enquanto que o procedural é opcional (ABNT, 2008-f). Já para os dispositivos *full-seg*, embora desde o início da televisão digital no Brasil as normas indicassem a obrigatoriedade dos dois subsistemas (ABNT, 2008-f), em decisão recente, com objetivo de agilizar a utilização da interatividade, o Fórum SBTVD optou em lançar o *middleware* Ginga em duas versões, a 1.0 e a 2.0. A versão 2.0 será a versão Ginga completa, incluindo os subsistemas Ginga-J e Ginga-NCL, enquanto que a versão 1.0 será lançada apenas com o subsistema declarativo.

Um limitante hoje para programação em Java para TV digital é a não existência de um ambiente de simulação e testes do Ginga-J. Portanto, neste momento, a programação para TV digital ainda está limitada em utilizar as linguagens NCL e Lua.

### 2.2.8 Usabilidade na TV Digital

Quando desenvolvidos aplicativos para a TV digital, deve-se levar em conta as diferenças em relação ao computador pessoal. Pode-se citar que um telespectador assiste TV geralmente a uma distância 7 a 8 vezes a altura da tela, enquanto que um usuário de computador de 50 a 75 centímetros do mesmo (SOARES, 2009).

Deve-se considerar também a maior diversidade de audiência, desde os jovens até os idosos, com uma parcela importante não tendo experiência com computadores (BBC, 2006). A BBC (2006) cita que um a cada trinta de seus telespectadores possuem algum tipo de deficiência visual. Portanto, fontes claras e grandes devem ser utilizadas, com cores que mostram um bom contraste entre o texto e o fundo, atendendo a todas as audiências (BBC, 2006).

Em relação à apresentação de textos na televisão, isso já é um desafio: os telespectadores não estão acostumados a ler blocos estáticos de texto na tela; em segundo lugar e a qualidade de exibição de imagens paradas na televisão é baixa (BBC, 2006).

Ainda em relação ao texto, os serviços de televisão interativa da BBC utilizam o tipo de fonte Tirésias, uma vez que seu alfabeto é bem impresso mesmo com pequenos tamanhos e quando as letras necessitam ser esticadas ou espremidas, devido a diferentes resoluções dos diversos aparelhos (BBC, 2006). Porém, ocorre que em certas circunstâncias – como títulos, cabeçalhos, elementos navegacionais e publicidade – que outras fontes melhoram a apresentação (onde cita a fonte *Gill Sans* para esse fim).

Porém, no caso de utilização de fontes diferentes, se sugere manter a apresentação limpa e simples, não colocando mais do que duas fontes distintas na apresentação (BBC, 2006). Becker et al (2006), mediante uma análise tipográfica das aplicações veiculadas pela programadora de satélite brasileira, chegou a conclusão que a Sky faz uso de tipos que muito se assemelham com a variante condensada da família de tipos *Frutiger*. A Figura 7 ilustra a comparação entre as três fontes citadas.



Figura 7. Famílias de tipos utilizados pelas emissoras BBCi e SKY (BECKER et al, 2006).

O SBTVD define, conforme descrito anteriormente, como obrigatório o suporte a fonte Tirésias para dispositivos *full-seg* e Verdana para dispositivos *one-seg* (ABNT, 2008-f), o que indica a tendência para utilização dessas fontes na televisão digital brasileira.

Segundo a BBC (2006), as seguintes regras podem melhorar a legibilidade do texto na televisão:

- O corpo do texto em geral não deve ser menor do que 24 pontos;
- Em qualquer circunstância nenhum texto deve ser menor do que 18 pontos;
- Texto com fonte clara em um fundo escuro é ligeiramente mais legível na tela;
- Texto na tela necessita de um espaçamento de entrelinhas maior do que em texto impresso;
- Um texto de tela inteira deve conter no máximo 90 palavras. Porém, quando acompanhado por um vídeo em tela cheia ou em um quarto de tela, o texto deve ser reduzido aproximadamente à metade dessa quantidade. Por outro lado, se o usuário escolher se aprofundar no serviço interativo e selecionar histórias, maiores quantidades de texto podem ser utilizadas;
- O texto deve ser quebrado em pequenos blocos que podem ser lidos quase instantaneamente.

Becker et al. (2006), em sua proposta, recomendam os seguintes tamanhos de fonte: 36 pontos para títulos; 20 pontos para menus; 22 pontos para texto e 18 pontos para legendas de botões. A Figura 8 apresenta um protótipo de portal que ilustra essa proposta.



Figura 8. Tamanhos de textos sugeridos para a interface (BECKER et al, 2006).

Em relação às imagens, o monitor dos computadores utiliza pixels que são quadrados; na tela da televisão, eles são um pouco retangulares, aproximadamente 1,067 vezes mais largos do que altos. Ou seja, as imagens são exibidas de maneira mais “esticada” horizontalmente quando se trata de aparelhos televisores. Para corrigir essa disparidade, a BBC (2006) recomenda que as imagens destinadas à televisão, inicialmente criadas em computador com resolução de 768 pixels de largura por 576 de altura, devem ser reduzidas horizontalmente para 720 pixels em largura. Esse procedimento indica que, quando uma imagem for transmitida para a televisão, a imagem será levemente esticada, sendo assim apresentada na proporção correta.

Em relação a cores, a BBC (2006) sugere não utilizar cores muito fortes para não provocar distorções, pois o contraste da televisão é muito maior do que nos computadores. Como proposta, o recomendável é não utilizar cor branca ou preta pura. Então, o aconselhável é ter no máximo para a cor branca, aproximadamente 95% (ou 240/240/240 em RGB) e para a cor preta 5% (16/16/16 em RGB).

Em relação à navegação, as recomendações mais relevantes tratam da utilização dos botões coloridos, também conhecidos como botões interativos (BBC, 2006):

- Utilizar Gill Sans *Bold*, 22 pontos para textos que funcionam como indicação para uma ação de botão;
- Posicionar botões de cor horizontalmente, preferivelmente na parte inferior da tela, e sempre na mesma ordem: vermelho, verde, amarelo, azul (da esquerda para direita);
- Manter sempre a mesma posição do botão colorido na tela, mesmo que alguns dos botões não sejam utilizados;
- Manter as funções dos botões coloridos consistentemente do início ao fim da aplicação.

A Figura 9 ilustra algumas dessas recomendações.

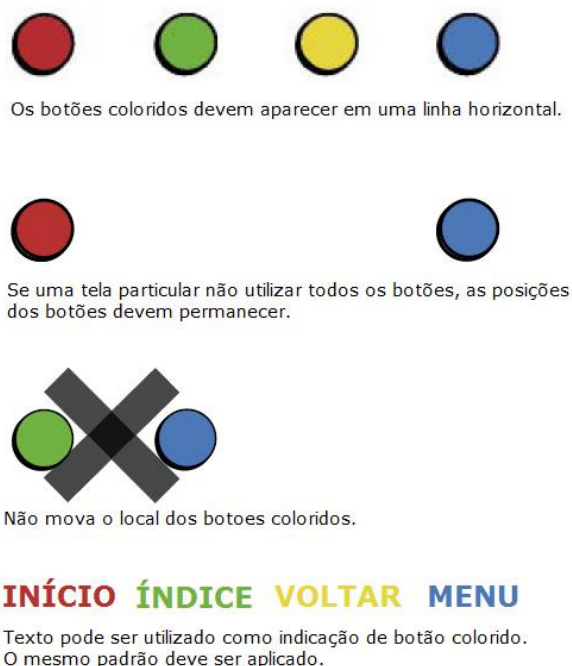


Figura 9. Utilização de botões coloridos. Adaptado de (BBC, 2006).

Com esse levantamento, realizado principalmente com base no *BBCi Designing For Interactive Television* (BBC, 2006) e nos estudos de Becker et al (2006), obteve-se uma série de recomendações sobre a interface dos aplicativos para TV digital. Essas recomendações foram aplicadas no desenvolvimento dos objetos de aprendizagem, demonstrados posteriormente.

## 2.3 Dispositivos Móveis

Dispositivos móveis se caracterizam por serem pequenos aparelhos capazes de oferecer a conveniência e a assistência de um computador convencional, porém, com limitações.

Como exemplo de dispositivos móveis comumente utilizados pode-se citar celulares, *smartphones* e PDAs (*Personal Digital Assistant*), bem como câmeras digitais, vídeo games portáteis, entre outros. Porém, para fins deste trabalho, refere-se a dispositivo móvel aquele capaz de interagir através de uma rede de telefonia móvel ou celular, focando nos aparelhos celulares e *smartphones*.

Não há uma clara distinção entre as categorias de telefones móveis. Geralmente refere-se ao celular como o aparelho com funcionalidades básicas de comunicação por voz e dados. Já um *smartphone*, normalmente é considerado como um celular com funcionalidades adicionais, como acesso a Internet, e-mails, teclado completo, etc. Por fim, um PDA possui adicionalmente serviços que facilitam tarefas corporativas, como agendas, calendários, além de tela sensível ao toque para anotações e outras atividades.

Em virtude do seu tamanho reduzido, os dispositivos móveis possuem algumas limitações a serem levadas em consideração ao se desenvolver aplicativos. Dentre elas pode-se citar:

- Consumo de energia: quanto menor o consumo da bateria, maior o tempo de uso do aparelho.

- Limite de processamento e de memória: recursos limitados pelo uso de energia e pelo tamanho reduzido.
- Menor banda disponível: qualidade e disponibilidade do sinal dependem da área de cobertura.
- Interação com o usuário limitada: dispositivos de interação, como teclado e tela, são menores.
- Compatibilidade entre diversas plataformas: *hardware* muito específico e APIs de desenvolvimento fechadas reduzem a compatibilidade entre as plataformas.

Embora diferentes aparelhos suportem diferentes linguagens de programação, as mais encontradas são o C++ (e em parte sua variante Symbian C++) e o Java.

Para acesso Web, diversos dispositivos possuem *mini-browsers* e acesso a Internet via redes de telefonia celular, que são capazes de apresentar componentes multimídia, como som, vídeo e imagem, além de pequenos aplicativos.

### 2.3.1 Áudio e Vídeo

O 3GPP SA WG4 (3GPP, 2010-c) é um subgrupo do 3GPP (3GPP, 2010-b) responsável pelas especificações para voz, áudio, vídeo, e *codecs* multimídia para dispositivos móveis.

Formato que aparecem amplamente disponíveis em um crescente número de aparelhos móveis são os formatos RA (*Real Audio*), para áudio, e RM (*Real Media*), para vídeo. Porém, esses são formatos proprietários da Real Networks (REAL NETWORKS, 2010). Os Real Players instalados em dispositivos Nokia (é o *player* utilizado por esse fabricante) suportam a decodificação de vídeo em formato *Real Video* 7 e 8, alguns também suportam os formatos 9 e 10 (NOKIA, 2010-b).

O formato de vídeo de padrão aberto, definido pela 3GPP, para uso em telefones móveis 3G (também pode ser usado em alguns telefones 2G e 4G) é o 3GP. Esse formato pode encapsular diversas codificações, como *streams* de vídeo H.263, MPEG-4, H.264 AVC, e *streams* de voz e áudio como o AMR-NB (*narrow-band speech*), AMR-WB (*wide-band speech*), AMR+ (*Extended wide-band*), MPEG-4 AAC, e AAC+ (3GPP, 2010-a).

A codificação de vídeo H.263 foi desenvolvida pela ITU (ITU, 2008) para comunicação a baixas taxas de transferência. Para a maioria dos serviços móveis de multimídia, o H.263 *Profile* 0, *Level* 10 (também conhecido como "H.263 *baseline*"), tem sido definido como um *codec* obrigatório. É também o *codec* principal suportado em *players* de vídeo da Nokia (NOKIA, 2010-b).

O MPEG-4 Visual (.MP4) é um *codec* de vídeo opcional em vários padrões recentes de multimídia. No entanto, é amplamente suportado nos mais recentes dispositivos Nokia disponível no mercado, adicionalmente ao H.263 (NOKIA, 2010-b).

Em relação ao áudio, os principais *codecs* de áudio usados em conteúdos para dispositivos móveis são o AAC e suas variações, o AMR, o RealAudio e o MP3 (NOKIA, 2010-c). As descrições sobre os *codecs* de áudio a seguir também foram realizadas com base em (NOKIA, 2010-c).

Um dos padrões mais avançados para codificação de fala é o AMR (*Adaptive Multi-rate*), devendo ser utilizada principalmente em mídias que não incluam conteúdos



musicais complexos, ruidosos ou de teor crítico para proporcionar melhor experiência auditiva. A sua banda de áudio limitada (3.5kHz) indica que as altas frequências não são reproduzidas perfeitamente e são filtradas por um *anti-aliasing*. Tipos de conteúdo provados funcionar satisfatoriamente com AMR incluem notícias, cobertura de esportes e música popular leve.

Já o AMR-WB representa o estado da arte em tecnologia de codificação de voz de banda larga com baixas taxas de *bits*. Este *codec* foi adotado pela 3GPP em dezembro de 2000 e pela ITU-T em Julho de 2001. AMR-WB usa o tipo MIME *audio/amr-wb* e sua extensão de arquivo é *awb*. A 3GPP tem diversas especificações disponíveis relacionadas a AMR-WB: 3GPP TS 26.190; 3GPP TS 26.201; 3GPP TS 26.174 e 3GPP TS 26.194.

O MP3 (MPEG-1 *Audio Layer 3*) é amplamente suportado nos aparelhos móveis mais recentes, com a extensão de arquivo *mp3*. É um formato de compressão com perdas, e aplica variadas técnicas para reduzir o tamanho dos dados de áudio, descartando informações baseado nas características do ouvido humano.

O AAC (MPEG-4 *Advanced Audio Coding*) é um formato sucessor do MP3 para codificação de áudio de médias a altas taxas de *bits*. Pode-se dizer que áudio codificado a 96 Kbps em AAC tem qualidade no mínimo igual ou superior a 128 Kbps em MP3. O AAC+ (também conhecido como *aacPlus*) é padronizado pelo MPEG sob o nome de HE-ACC (*High Efficiency AAC*). Um *stream* AAC+ a 48 kbps é considerado ter maior qualidade do que 128 Kbps em MP3. O eAAC+ (*enhanced AAC+*) melhora ainda mais o desempenho do *codec* a menores taxas de *bits*.

### 2.3.2 Imagens

No que concerne ao suporte a formatos de imagens em dispositivos móveis, os formatos GIF, JPEG e PNG, já amplamente difundidos e utilizados em páginas dos *browsers* de computadores pessoais, também são suportados pela grande maioria dos dispositivos móveis atuais. O suporte a esses formatos pode ser observado nas configurações dos celulares no site de seus fabricantes (NOKIA, 2010-a) (MOTOROLA, 2010) (SAMSUNG, 2010) (SONY ERICSSON, 2010) (LG ELECTRONICS, 2010). Adicionalmente, o uso dos formatos GIF 89a e JPEG em conteúdos para Web móvel é uma recomendação W3C (W3C, 2008-b).

Além dos formatos gráficos citados acima, ainda existe outro formato otimizado para dispositivos móveis chamado WBMP (Wireless Bitmap). O WAE<sup>10</sup> define um formato para imagens gráficas chamado Wireless Bitmap (WBMP), ele foi projetado para atender vários requisitos concorrentes, incluindo suporte a múltiplas profundidades de pixels, suporte a tabelas de espaço de cores, baixa codificação e baixa demanda de CPU e RAM. O formato WBMP permite que informações gráficas sejam enviadas para uma variedade de aparelhos portáteis independente do terminal, pois descreve apenas informação gráfica (WAP FORUM, 2001).

---

<sup>10</sup> *Wireless Application Environment* (WAE) é parte do esforço do WAP Forum para especificar um *framework* de aplicação para terminais sem fio, como telefones celulares, *paggers* e PDAs. O *framework* amplia e impulsiona outras tecnologias WAP, incluindo sessão, protocolos de transporte e segurança, bem como tecnologias de Internet como XML, URLs, scripts, e diversos tipos de mídia. O esforço permite aos operadores, fabricantes e desenvolvedores de conteúdo enfrentar de forma rápida e flexível os desafios na construção de serviços e implementações avançadas e diferenciadas (WAP FORUM, 2001).

### 2.3.3 Extensible HyperText Markup Language (XHTML)

A fim de permitir aos fornecedores de conteúdo uma visão consistente e comum de um cenário de uso básico para Web no celular, o *Mobile Web Best Practices Working Group*<sup>11</sup> (BPWG) definiu o Contexto Padrão de Entrega. Esse determina as especificações mínimas necessárias do contexto de entrega<sup>12</sup> para proporcionar ao usuário uma experiência razoável na Web. A Tabela 2.2 descreve as definições do Contexto Padrão de Entrega.

Tabela 2.2: Definições do Contexto Padrão de Entrega.

<b>Largura utilizável de tela</b>	Mínimo de 120 pixels
<b>Suporte a Linguagem de Marcação</b>	XHTML <i>Basic</i> 1.1 entregue com content type <i>application/xhtml+xml</i>
<b>Codificação de caracteres</b>	UTF-8
<b>Suporte a Formato de Imagem</b>	JPEG, GIF 89a
<b>Tamanho Máximo Total da Página</b>	20 kilobytes
<b>Cores</b>	Mínimo de 256 cores
<b>Suporte a Folha de Estilos</b>	CSS Level 1. Além disso, regra CSS Level 2 @media em conjunto com os tipos de mídia <i>handheld</i> e <i>all</i> .
<b>HTTP</b>	HTTP/1.0 ou mais recente
<b>Script</b>	Sem suporte a linguagens <i>script</i> no lado cliente

Fonte: W3C, 2008-b.

O BPWG sugere que, se possível, seja usado um mecanismo de adaptação para aproveitar os recursos adicionais de algum contexto de entrega mais capacitado.

### 2.3.4 Cascading Style Sheet (CSS)

Também existem versões de CSS específicas para as plataformas móveis. Nos dispositivos móveis são utilizados os seguintes perfis de CSS: CSS Mobile Profile 2.0 (W3C, 2008-a) e Wireless CSS 1.2 (OMA, 2008).

Essas especificações definem um subconjunto do CSS 2.1, utilizado na Web, que é para ser considerada a base para interoperabilidade entre implementações de CSS em dispositivos restritos (como telefones celulares). O objetivo não é gerar um perfil incompatível com a especificação completa, mas assegurar que todas as plataformas com restrições de recursos e que não podem dar suporte a especificação completa, implementem este subconjunto comum, ficando não apenas compatíveis entre si, mas também com a especificação CSS completa (W3C, 2008-a).

<sup>11</sup> O Mobile Web BPWG é um grupo criado em 2005 pela W3C como parte da iniciativa para web móvel. Sua função é desenvolver um conjunto de melhores práticas, técnicas e materiais associados de apoio ao desenvolvimento de web sites que forneçam uma experiência adequada ao usuário em dispositivos móveis.

<sup>12</sup> Contexto de entrega (*Delivery Context*) define um conjunto de atributos que caracterizam as capacidades do mecanismo de acesso, as preferências do usuário e outros aspectos do contexto no qual uma página Web deverá ser entregue.

Adicionalmente, essa especificação está de acordo com a especificação OMA Wireless CSS 1.1, ao mesmo tempo em que a OMA está realizando trabalho de alinhamento na OMA Wireless CSS 1.3, para compatibilizar o CSS Mobile Profile 2.0 e o OMA Wireless CSS 1.2 (W3C, 2008-a).

### 2.3.5 Linguagem Script

O perfil de linguagem *script* recomendado para dispositivos móveis é o ECMAScript – Mobile Profile (ESMP). O ESMP ou OMA *Wireless Markup Scripting Language* é fortemente baseado no ECMA-262 (descrito na seção referente à plataforma de computadores móveis). Esse mesmo grupo também criou um perfil para a linguagem ECMAScript para dispositivos móveis, o ECMA-327. A linguagem foi modificada para dar melhor suporte a comunicação em baixas larguras de banda e *thin-clients*. É recomendada a utilização do ESMP junto com perfil de XHTML para celular, sendo esse conhecido como XHTML-MP (OMA, 2006).

### 2.3.6 Java

*Java Micro Edition* (JavaME) é a especificação de um subconjunto da plataforma Java destinada a sistemas embarcados. Existem dois tipos de configurações de Java ME: *Connected Limited Device Configuration* (CLDC) e *Connected Device Configuration* (CDC). O CDC é usado para dispositivos com maior capacidade computacional enquanto que o CLDC destina-se aos com menor capacidade. Estas configurações definem um ambiente de execução JRE e as classes básicas, que operam sobre cada dispositivo. Para dispositivos móveis como celulares é usado o CLDC, disponível nas versões 1.0 e 1.1.

Cada dispositivo Java ME implementa um perfil (*profile*). Um perfil consiste em um conjunto de classes que, em conjunto com uma configuração, possibilita aos desenvolvedores de *software* implementar aplicações de acordo com as características do dispositivo. O *Mobile Information Device Profile* (MIDP) (SUN MICROSYSTEMS, 2010-e) é o *profile* de CLDC projetado para dispositivos móveis como aparelhos celulares e PDAs. Atualmente está disponível nas versões MIDP 1.0, 2.0 e 2.1. Praticamente todos os telefones celulares atualmente vêm com uma implementação de MIDP, de forma que ele se tornou o padrão de fato para jogos de celulares.

É possível ainda estender a plataforma JavaME com pacotes opcionais. Esses pacotes oferecem APIs padronizadas permitindo o uso de outras tecnologias como *wireless*, multimídia, *bluetooth*. A API MMAPI, por exemplo, é utilizada para estender a funcionalidade da plataforma provendo áudio, vídeo e outros suportes multimídias baseados em tempo (SUN MICROSYSTEMS, 2010-f).

### 2.3.7 Flash

*Adobe Flash Lite* (ADOBE, 2010-b) é uma versão mais simples (sem alguns recursos) do Adobe Flash Player (ADOBE, 2010-a) que permite aos usuários de dispositivos móveis visualizarem conteúdos multimídia e aplicações desenvolvidos com as ferramentas Adobe Flash.

Existem diferentes versões *Flash Lite* atualmente disponíveis para aparelhos celulares. As versões *Macromedia Flash Lite* 1.0 e 1.1 são baseadas no Flash Player 4. Enquanto *Flash Lite* 2.0 e 2.1 são baseados no *Macromedia Flash Player* 7 e suportam a maioria, das funcionalidades do Flash Player 7. *Flash Lite* 2.x também inclui recursos

específicos para desenvolvimento móvel que não estão disponíveis no Flash Player 7. Por exemplo, com Flash *Lite* 2.x, é possível carregar tipos de mídia específicos do dispositivo (imagens, sons, vídeos) que não são suportados nativamente pelo Flash *Lite*. Flash *Lite* 2.x também inclui funcionalidades de integração com o dispositivo, tais como a capacidade de fazer chamadas telefônicas e enviar mensagens de texto.

O Flash *Lite* 3.0, também baseado em Flash Player 7, introduz suporte a Flash Vídeo. Ele também inclui um novo modelo de segurança espelhado no modelo utilizado no Flash Player 8.

Flash *Lite* está instalado em uma variedade de dispositivos móveis. Cada instalação suporta um ou mais modos de aplicação ou tipos de conteúdo. Por exemplo, alguns dispositivos utilizam o Flash *Lite* para habilitar *screen savers* animados ou toques de chamada. Outros dispositivos utilizam o Flash *Lite* para renderizar conteúdo Flash embutido em páginas da Web móvel (ADOBE, 2010-b).

### 2.3.8 Usabilidade em Dispositivos Móveis

Não existe nenhuma recomendação oficial de padrão para fontes a serem utilizadas em dispositivo móveis, nem mesmo no W3C *Mobile Web Best Practices* (W3C, 2008-b). No entanto, em (NOKIA, 2007), são fornecidas algumas recomendações de usabilidade para os navegadores Web de seus dispositivos.

Em relação à fonte do texto, cada *browser* utiliza uma fonte própria. Por esse motivo, não é necessário e nem adiantará se preocupar com a fonte, pois, independentemente do que for especificado para o HTML, os *mini-browsers* utilizarão seu próprio estilo de fonte.

Em relação aos tamanhos, para um padrão de tela pequeno, como o QVGA (um quarto de VGA, 240 x 320), os seguintes tamanhos de fonte são recomendados:

- Título 1: 18px, negrito
- Título 2: 16px, negrito
- Título 3: 14px, negrito
- Título 4: 12px, negrito
- Texto normal: 12px, normal

## 3 PADRÕES DE METADADOS

Diversos padrões de metadados foram propostos ao longo dos últimos anos com objetivos distintos. Este capítulo apresenta um estudo geral destes padrões de metadados, iniciando pelo padrão Dublin Core, proposto como padrão básico para a Web, os padrões existentes para TV digital e os padrões educacionais.

### 3.1 *Dublin Core Metadata Initiative* (DCMI)

O *Dublin Core Metadata Initiative*, inicialmente definido na RFC 5013 para Web (KUNZE, BAKER, 2007), é um vocabulário contendo quinze propriedades para descrição de recursos. O termo Dublin deriva da sua origem em um *workshop* efetuado em Dublin, em 1995, onde se reuniram bibliotecários, pesquisadores de bibliotecas digitais, especialistas em conteúdo e linguagens Web. O termo *core* foi utilizado porque seus elementos são julgados como mínimo imprescindível para descrever recursos e que, ao mesmo tempo, são genéricos e utilizáveis para descrever uma grande variedade de recursos.

Esse núcleo de metadados é descrito a seguir (DCMI, 2009-a):

1. *Title*: título do recurso;
2. *Creator*: normalmente o nome da entidade responsável;
3. *Subject*: normalmente representado por palavras-chave ou frases-chave;
4. *Description*: pode conter resumo, índice, figuras, texto explicativo, etc.;
5. *Publisher*: entidade responsável por disponibilizar o recurso;
6. *Contributor*: alguém que contribuiu para o recurso, porém, não é autor;
7. *Date*: data ou período de tempo associado ao evento. Sugere-se que seja codificado no formato W3CDTF ou ISO 8601. Por exemplo, para a data completa mais horas e minutos o formato seria: YYYY-MM-DDThh:mmTZD, por exemplo, 2008-04-03T10:53+03:00. TZD indica o fuso horário da localidade;
8. *Type*: natureza do recurso. Recomenda-se um vocabulário padrão, como o *DCMI Type Vocabulary* (DCMI, 2009-b), onde os tipos definidos são: *Collection*, *Dataset*, *Event*, *Image*, *InteractiveResource*, *MovingImage*, *PhysicalObject*, *Service*, *Software*, *Sound*, *StillImage* e *Text*;
9. *Format*: formato do arquivo, meio físico, dimensões, tamanho do arquivo, duração. Recomenda-se limitar-se aos tipos MIME (IANA, 2007), que

possui os seguintes formatos definidos: *application, audio, example, image, message, model, multipart, text, video*;

10. *Identifier*: identificador único do recurso. Sugere-se associação a uma string utilizando um mecanismo de identificação formal, como URLs (*Uniform Resource Locator*);
11. *Source*: recurso relacionado de onde este recurso é derivado (quando é uma parte de um todo). Recomenda-se utilizar mecanismo formal, como URLs;
12. *Language*: linguagem, conforme RFC 4646 (PHILLIPS e DAVIS, 2006);
13. *Relation*: recurso relacionado. Recomenda-se utilizar mecanismo formal, tipo URLs;
14. *Coverage*: tópico espacial ou temporal do recurso, como um nome de lugar, coordenadas geográficas, período, data. Recomenda-se usar o *Getty Thesaurus of Geographic Names* (TRUST, 2004), conforme exemplo da Figura 10.
15. *Rights*: Informações sobre os detentores de direitos autorais associados com o recurso.

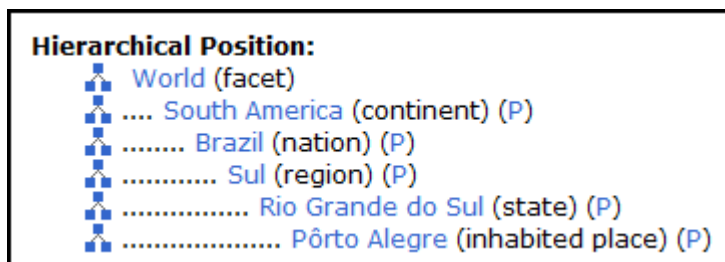


Figura 10. Hierarquia de uma pesquisa no TGN para “Porto Alegre” (TRUST, 2010).

### 3.2 MPEG-7

O MPEG-7, formalmente chamado de *Multimedia Content Description Interface*, é um padrão ISO/IEC voltado a arquivos multimídia, desenvolvido pelo *Moving Picture Experts Group* (MPEG), no ano de 2001.

O objetivo do MPEG-7 (2004) é fornecer uma descrição de conteúdo áudio-visual, garantindo a interoperabilidade entre aplicativos multimídia em busca, indexação, filtragem e acesso de conteúdo. Isso possibilita que aplicativos diversos possam trabalhar com metadados multimídia. O diferencial desse padrão, comparando-se a outros, é a flexibilidade em relação ao que pode ser descrito, permitindo tanto a descrição de informações semânticas e complexas, como estruturas mais simples.

O MPEG-7 é um padrão aberto baseado no formato XML (*Extensible Markup Language*), permitindo que aperfeiçoamentos sejam realizados continuamente. Conforme ilustrado na Figura 11, apresenta como componentes básicos o *Description Tools* e o *Description Definition Language* (DDL).

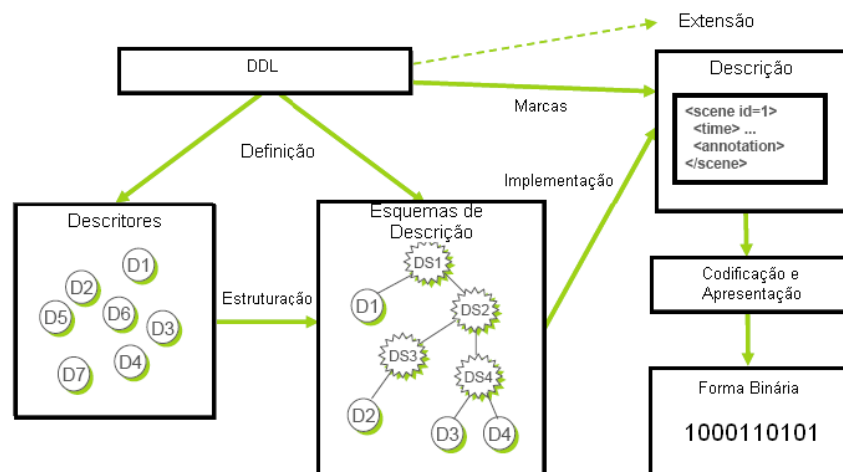


Figura 11. Elementos Principais das Descrições MPEG-7 (CHANG, 2001).

O *Description Tools* é um sistema de ferramentas que serve para criar descrições, estando esse dividido em:

- *Descriptor* (descritores): representam uma funcionalidade e definem a sintaxe e semântica para a sua representação (elemento do metadado), possibilitando a descrição de áudios, vídeos e imagens, além de descrições abstratas de conteúdo multimídia e descrição do processo de criação e autoria, gerenciamento e classificação de conteúdos, além de representar as preferências do usuário;
- *Description Schemes* (DS): especificam a estrutura e a semântica dos relacionamentos entre seus componentes, podendo ser descritores ou esquemas como, por exemplo, um vídeo armazenado como cenas, com descritores textuais, de áudio e cores.

O *Description Definition Language* permite a criação de novos esquemas e descritores, bem como a extensão e modificação de esquemas (ALVES, 2006).

Formalmente, o padrão MPEG-7 divide-se em:

- Parte 1 – Sistemas (*Systems*): descreve a arquitetura dos terminais MPEG-7, além de especificar o formato de codificação para transporte e armazenamento de descritores;
- Parte 2 – Linguagem de Definição de Descrição (*Description Definition Language*): linguagem para definição da sintaxe dos descritores MPEG-7 e definição de novos *description schemes*. A DDL é composta pelos seguintes componentes: a) componentes estruturais de esquema XML, b) tipos de dados de esquemas XML, c) extensões específicas MPEG-7;
- Parte 3 – Visual: especifica descritores e esquemas de descrição para dados visuais, tais como características de cor, textura, forma, movimento, localização e reconhecimento de faces;
- Parte 4 – Áudio: especifica ferramentas de descrição para tratar quatro classes de sinais de áudio (somente música, somente fala, somente efeitos sonoros e trilhas sonoras);

- Parte 5 – Esquemas de Descrição Multimídia (*Multimedia Description Schemes*): trata de características genéricas de descrições multimídia;
- Parte 6 – Software de Referência (*Reference Software*): software que implementa partes relevantes do padrão MPEG-7 em situação normativa;
- Parte 7 – Testes de Compatibilidade (*Conformance Testing*): guias e procedimentos para testar a conformidade das implementações do MPEG-7;
- Parte 8 – Extração e Uso de Descrições (*Extraction: and use of Descriptions*): material informativo em forma de relatórios técnicos (*technical report*) sobre extração e utilização de alguns dos conjuntos de descritores especificados pelo padrão (*description tools*);
- Parte 9 – Perfis (*Profiles*): fornecem diretrizes e perfis dos padrões;
- Parte 10 – Definição de Esquema (*Schema Definition*): especifica o esquema, utilizando a linguagem de definição da descrição.

Conforme a Figura 12, a descrição pode ser obtida do conteúdo de mídia por extração manual ou automática e os metadados são armazenados para futuro uso. No cenário “recebe”, um conjunto de descrições que está de acordo com o pedido do usuário, é enviado a ele para análise. No cenário “envia”, um agente automático pode filtrar as descrições e executar ações programadas, como mudar de canal de TV ou gravar uma determinada transmissão (CHANG, 2001).

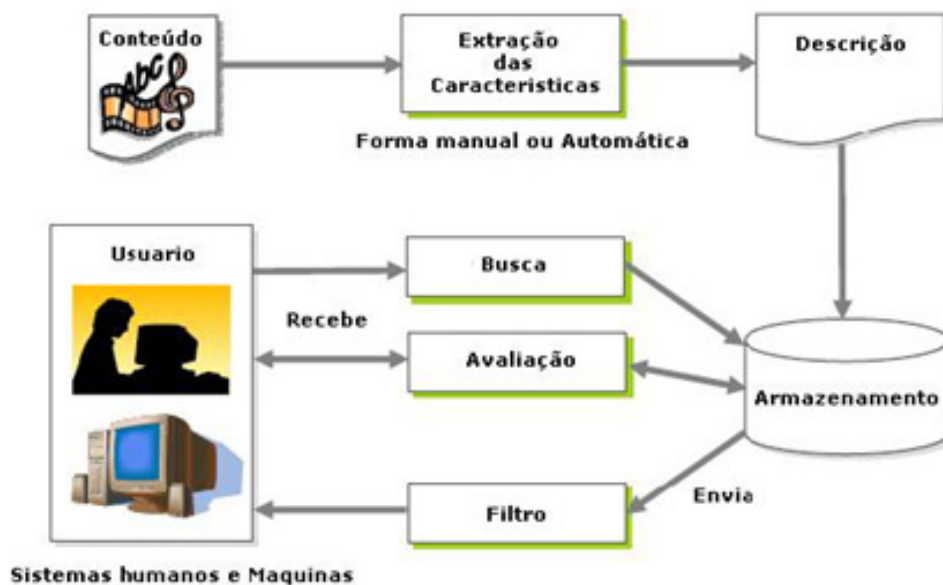


Figura 12. Exemplo de cadeia MPEG-7 (CHANG, 2001).

Uma vez que a ênfase do padrão está no fornecimento de soluções para descrição de conteúdos audiovisuais, o desenvolvimento de algoritmos e ferramentas de busca ou para extração automática e semi-automática de características encontram-se fora dos objetivos de MPEG-7. A forma como os dados MPEG-7 serão utilizados para responder as perguntas do usuário também não faz parte dos objetivos desse padrão, sendo de responsabilidade dos mecanismos de busca associar os dados da pesquisa à descrição MPEG-7 de um material audiovisual (CHANG, 2001).



Como os descritores MPEG-7 independem do modo como o conteúdo descrito foi codificado ou armazenado, uma das vantagens é a possibilidade de criação de uma descrição tanto para um filme, quanto para uma figura impressa em um papel, ou seja, permite diferentes granularidades e o seu uso em uma grande variedade de áreas, como em acervos multimídia e TV digital.

### 3.3 MPEG-21

O formato MPEG-21, da ISO/IEC, define um padrão onde os diferentes elementos formam uma infra-estrutura de entrega e consumo de conteúdos de mídia, constituída por 17 partes, combinadas em quatro grupos, conforme a Figura 13.

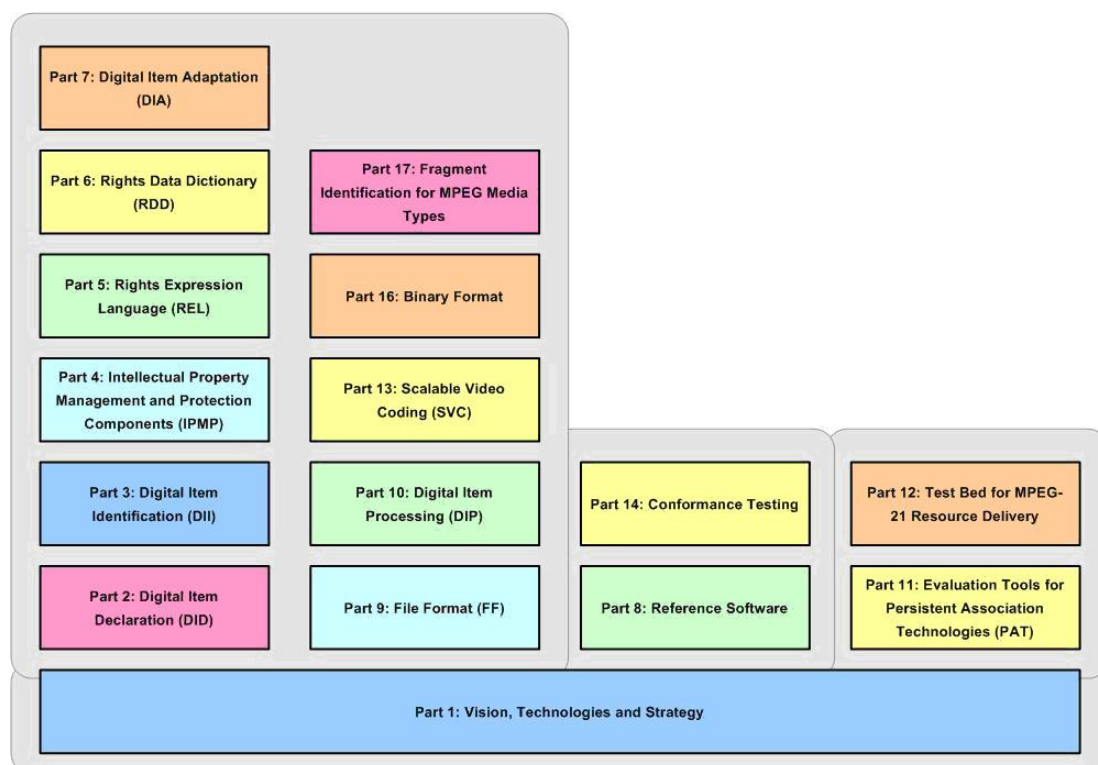


Figura 13. Partes relacionadas ao MPEG-21 (ITEC, 2008).

O primeiro grupo (partes 1 até 7) constitui a parte do *framework* dedicada à identificação, representação e controle de propriedade intelectual dos itens digitais. O segundo grupo (partes 9, 10, 13, 16 e 17) constitui a parte do *framework* preocupada com as questões de processamento e manipulação dos itens digitais. O terceiro grupo (partes 8 e 14) constitui a parte do *framework* voltada aos testes e validações da conformidade dos itens digitais com as especificações criadas pelo primeiro grupo. Por fim, o quarto grupo (partes 11 e 12) constitui a parte do *framework* envolvida com a distribuição e persistência dos itens digitais.

O primeiro grupo se divide em:

- *Part 1: Vision, Technologies and Strategy*: define um *framework* multimídia que permite o uso transparente e interoperável de recursos multimídia através de uma larga rede de computadores e outros dispositivos de acesso a dados, considerando as permissões, direitos e propriedade intelectual. Para tanto, estabelece padrões e estratégias de integração de tecnologias para a criação,

gerência, transporte, manipulação, distribuição e consumo de itens digitais. Esse *framework* é baseado em dois conceitos essenciais: a definição de uma unidade fundamental de distribuição e transação, chamada Item Digital, e o conceito de Usuários que interagem com os Itens Digitais;

- *Part 2: Digital Item Declaration (DID)*: define um conjunto de termos e conceitos abstratos que especificam um Item Digital. Estabelece os seguintes itens: a) *model*: é o conjunto de termos que define um Item Digital; b) *representation*: descrição da sintaxe e semântica de cada DID através do formato XML; c) *schema*: um esquema XML descrevendo a gramática do DID;
- *Part 3: Digital Item Identification (DII)*: provê um mecanismo para identificar diferentes tipos de itens digitais e realizar a identificação única desses, do IP relacionado e do esquema descritivo de cada item digital. Além disso, possibilita usar os identificadores únicos para relacionar os itens digitais com outras informações, tais como metadados;
- *Part 4: Intellectual Property Management and Protection Components (IPMP)*: fornece os mecanismos necessários para a proteção dos direitos autorais, a saber: a) *IPMP Digital Item Declaration Language*: uma norma alternativa para a representação de partes do item digital que precisam de proteção autoral; b) *IPMP Information schemas*: define um esquema que contém informações a respeito da proteção do conteúdo de um item digital, incluindo ferramentas, mecanismos e licenças;
- *Part 5: Rights Expression Language (REL)*: define uma linguagem baseada nos termos da RDD (*Rights Data Dictionary*), para declarar direitos e permissões sobre itens digitais;
- *Part 6: Rights Data Dictionary (RDD)*: compreende um conjunto claro, coerente, estruturado, integrado e único de termos que apóiam a máquina de expressões regulares do REL;
- *Part 7: Digital Item Adaptation (DIA)*: especifica um conjunto de ferramentas que garante a adaptação dos itens digitais. Esses mecanismos ajudam a manter informações a respeito de características do usuário (necessidades especiais - acessibilidade, preferências de exibição de um determinado tipo de mídia, etc.), da capacidade do terminal (capacidade de codificar/decodificar certos tipos de mídia, tipo de sistema operacional, protocolos de comunicação, etc.), características da conexão (tamanho de banda, velocidade de acesso, etc.), entre outros.

### 3.4 IEEE Learning Object Metadata (IEEE-LOM)

Oficialmente conhecido como IEEE 1484.12.1-2002 *Standard for Learning Object Metadata*, o IEEE-LOM é um padrão aberto e internacionalmente reconhecido para facilitar a busca, avaliação, construção e uso de objetos de aprendizagem (OA), provendo um modelo de dados normalmente codificado em XML (IEEE, 2002).

O objetivo desse padrão é especificar a sintaxe e a semântica de informações (metadados) a respeito de um OA. Essas especificações permitem catalogar os materiais educacionais (através de seus metadados), tendo em conta a diversidade de contextos

culturais e linguísticos da criação e reuso dos OAs. Dessa maneira, procura-se garantir modos eficientes de identificação, (re) utilização, gerenciamento, interoperabilidade, compartilhamento, integração e recuperação de objetos de aprendizagem.

O padrão IEEE-LOM também pode ser visto como uma especificação de um cabeçalho que fornece informações sobre o objeto de aprendizagem. Os elementos de dados que compõem esse cabeçalho são os metadados a respeito do OA. Assim, esse padrão não interfere no conteúdo ou nas regras dos objetos de aprendizagem, uma vez que apenas agrupa metadados. Por isso, alcançou amplo uso em EAD, como, por exemplo, nos repositórios CAREO (australiano), ABED e RIVED (brasileiros) e com diversas iniciativas de padronização, como a certificação ADL/SCORM (ADL, 2004).

Na prática, a norma IEEE-LOM define uma biblioteca de metadados que podem ser combinados livremente para criar o cabeçalho de informações do OA, visando definir um conjunto mínimo de atributos necessários para o correto gerenciamento, localização e avaliação de objetos de aprendizagem.

Os metadados usados para descrever o objeto de aprendizagem incluem informações pedagógicas (como forma de ensino, nível de escolaridade e pré-requisitos), além de manter metadados sobre tipo de objeto, autor, proprietário, termos de distribuição e formato, apresentando forte preocupação com as questões de segurança, privacidade, uso e comercialização (IEEE, 2002).

A estrutura básica dos metadados é classificada nos seguintes grupos (IEEE, 2002):

- Geral: contém informações sobre o objeto como um todo;
- Ciclo de vida: contém metadados sobre a evolução do objeto;
- Meta-metadados: informa sobre os metadados que descrevem o objeto;
- Técnico: apresenta descrição de características e requisitos técnicos;
- Educacional: contém atributos educacionais e pedagógicos;
- Direitos: descreve direitos relacionados à propriedade intelectual e condições de uso;
- Relação: identifica objetos relacionados;
- Notação: contém comentários e a data, além do autor do comentário;
- Classificação: descreve o objeto em relação a um sistema de classificação particular.

### 3.5 SCORM

O SCORM (*Sharable Content Object Reference Model*) (ADL, 2004) é um modelo de referência que define especificamente sistemas de gerência de aprendizado (LMS – *Learning Management Systems*) e conteúdo educacional de forma que eles interoperem entre diferentes sistemas compatíveis.

Basicamente, trata de dois grandes elementos: empacotamento de conteúdo e troca de dados em tempo de execução. O primeiro trata da forma que o conteúdo deve ser entregue no sentido físico, contendo no seu núcleo um documento chamado *imsmanifest*, que contém toda informação necessária para o LMS importar e lançar o

conteúdo sem intervenção humana. O segundo trata da troca de dados, especificando como o conteúdo interage com o LMS em tempo de execução.

As principais diretivas são o *find*, para encontrar o LMS, e outras para comunicação (*set* e *get*).

### 3.6 Tabelas SI/PSI

Os metadados mais básicos para TV digital são compostos das tabelas SI (*Service Information*) e PSI (*Program Specific Information*) (ABNT, 2008-e), que permitem obter informações sobre algumas informações elementares da programação, como o horário, duração, título, classificação de faixa etária, um pequeno resumo dos programas, e pouco mais que isso.

Essas tabelas são parte do padrão de transporte MPEG-2 e contêm os metadados de descrição das programações enviadas pelas emissoras. Atualmente, o MPEG-2 é o padrão adotado pelos principais sistemas de TV digital em operação no mundo, como o DVB (padrão europeu), ATSC (padrão americano) e ISDB (padrão japonês, adotado também no Brasil).

Para entender como funcionam as tabelas SI, primeiramente é necessário compreender como a programação vai da emissora até o receptor de TV digital. Um programa de televisão é formado por um ou mais arquivos de áudio, vídeo e dados. Cada um destes arquivos é chamado de fluxo elementar (*Elementary Stream - ES*) (CPqD, 2006). Por sua vez, os componentes correspondentes aos fluxos elementares de informação de um serviço (programa de TV) são acomodados em pacotes (*Packetized Elementary Stream Packets - PES*) para então serem multiplexados em um feixe de transporte (*Transport Stream - TS*) MPEG-2, conforme pode ser visto na Figura 14.

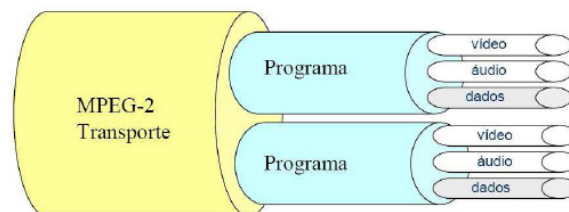


Figura 14. Fluxos elementares formando um feixe de transporte MPEG-2 (MONTEZ, 2005).

O feixe de transporte é composto por uma sequência contínua de pacotes denominados *TS packets*. A Figura 15 ilustra um exemplo de feixe de transporte MPEG-2.

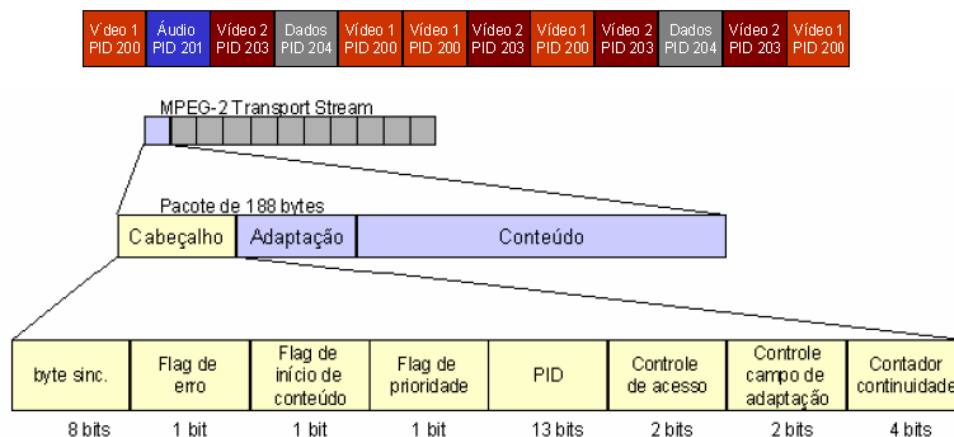


Figura 15. Exemplo de um MPEG-2 *Transport Stream* (ANDREATA, 2006).

Além dos fluxos elementares de áudio, vídeo e dados citados anteriormente, é necessário que um conjunto básico de informações de programa seja enviado para descrever o conteúdo transportado. A esses metadados dá-se o nome de tabelas SI (CPqD, 2006). Com os dados transmitidos pelas tabelas SI, torna-se possível, através de um receptor digital de televisão terrestre, a seleção de canais e eventos existentes, podendo ser utilizadas, por exemplo, como base para criação da grade de programação eletrônica (*Electronic Program Guide* - EPG) (ABNT, 2008-d), como a da Figura 16.



Figura 16. Exemplo de um EPG (NEBULA, 2010).

As tabelas SI são compostas por um conjunto de tabelas hierarquicamente associadas. As tabelas básicas são chamadas de tabelas PSI/MPEG-2 (*Program Specific Information* – MPEG-2) (ABNT, 2008-d), sendo elas: PAT (*Program Association Table*), PMT (*Program Map Table*) e CAT (*Conditional Access Table*). A Figura 17 ilustra as tabelas e seus relacionamentos.

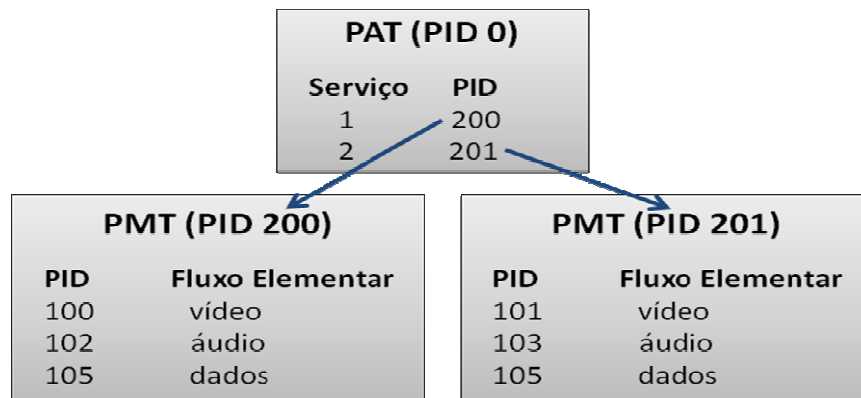


Figura 17. Serviços descritos através de conjunto de tabelas PSI. Adaptado de (MONTEZ, 2005).

A tabela PAT (identificada sempre pelo PID – *packet identifier* – 0x0000) contém os indicadores dos pacotes TS que carregam as tabelas PMT dos programas. A tabela PMT contém informações de acesso condicional comum aos programas e providencia o apontamento para os elementos daquele programa, ou seja, contém o identificador dos pacotes TS que carregam os sinais codificados (áudio, vídeo e dados) que formam cada um dos programas de radiodifusão (ABNT, 2008-c). A Figura 18 exemplifica esse relacionamento.

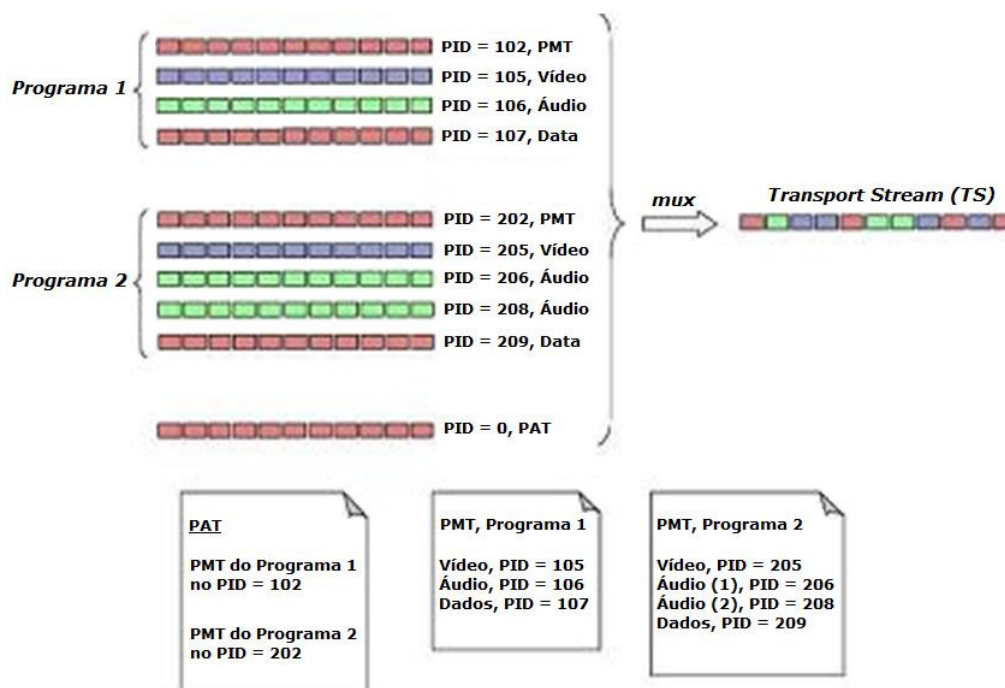


Figura 18. Relacionamento entre a *Transport Stream* e as tabelas SI (ANDREATA, 2006).

Já a tabela CAT fornece uma associação entre o sistema de acesso condicional e outra tabela, chamada EMM (*Entitlement Management Messages*), responsável pelas regras de acesso condicional individual (ABNT, 2008-c).

Em relação ao grupo das tabelas SI, a tabela EIT fornece a informação básica da programação que está sendo transmitida (ABNT, 2008-e). Essa tabela contém os descritores de título, descrição completa, descrição resumida, hora de início, duração,

categoria do conteúdo, classificação indicativa, controle de cópia e informação a respeito dos componentes.

A tabela de informação de um evento local (*Local Event Table* - LIT) inclui informações relacionadas ao evento local (evento de segmentação do programa) como nome, tempo de início e duração desse evento. Para relacionar o grupo de evento (programa) ao evento local, é utilizada a tabela de relação de eventos (*Event Relation Table* - ERT) (ABNT, 2008-e).

Outras duas tabelas importantes são: tabela de data e horário (*Time and Date Table* - TDT) e a tabela de diferença de horário (*Time Offset Table* - TOT). A TDT é responsável por levar a informação de horário e a informação de data. Já a tabela TOT, além de data e horário, contém também informações sobre a diferença de fuso horário (ABNT, 2008-e).

### 3.7 TV-ANYTIME

O TV-Anytime é uma associação de organizações formada em uma reunião realizada em *Newport Beach*, Califórnia, EUA, no ano de 1999 (LUGMAYR, NIIRANEN, KALLI, 2004), visto a necessidade de acolher os diversos e novos serviços que o atual mercado tecnológico exige, como consumidores mais autônomos e no controle de captar, armazenar, verificar e distribuir conteúdo para suas próprias redes pessoais e outros ambientes digitais, além do compartilhamento com outras pessoas. Jogos, informações e pacotes educativos e/ou recreativos, t-comércio, e/ou serviços utilitários (como bancos, lojas e aplicações financeiras) são outras opções. Junto a tudo isso, os provedores estão preocupados em saber qual conteúdo é mais relevante para seus usuários e a capacidade de armazenar, monitorar, mover e redistribuir esse conteúdo, de forma gratuita ou licenciada. Essas especificações prevêm a procura, seleção, aquisição e uso legal de conteúdo particular armazenado local e/ou remotamente vindo por *broadcast* ou de serviços online (pelo canal de interatividade) (TVA, 2001).

O TV-Anytime se divide em duas fases (TVA, 2001):

- Fase 1: destina-se à busca de conteúdos de áudio e vídeo, captura e reprodução desses conteúdos, além de permitir a segmentação e indexação.
- Fase 2: destina-se a especificar padrões mais abertos para sistemas integrados e interoperáveis, possibilitando a adição de outros tipos de conteúdo (jogos, páginas Web, programas de TV, gráficos, etc.), juntamente com conteúdos de áudio e vídeo. Inclui o empacotamento, segmentação, sincronismo e programação remota. Além disso, nessa fase, são incluídas disposições relativas à proteção dos direitos autorais. Para tanto, é exigida a descrição de novos tipos de metadados que possam ser transferidos entre todos os dispositivos de informações.

Dentre os objetivos, o Fórum TV-Anytime estabeleceu quatro como fundamentais (TVA, 2003-a):

- Definir especificações que permitam aplicações que explorem o armazenamento local de mídias pessoais persistentes nas plataformas eletrônicas dos consumidores;

- Considerar-se como rede independente no que se refere ao conteúdo dos meios de entrega de equipamentos eletrônicos, incluindo diferentes mecanismos de entrega de TV digital (por exemplo, ATSC, DVB, ISDB e outros), bem como Internet e um aprimorado sistema de TV;
- Desenvolver especificações para sistemas operáveis e integrados, a partir de criadores e/ou fornecedores de conteúdo, aos consumidores;
- Especificar as estruturas de segurança para proteger os interesses de todas as partes envolvidas.

Quando o conteúdo é protegido, os direitos de uso e transmissão podem estar restritos a grupos de usuários e domínios específicos, limites regionais, nacionais e internacionais, e até a alguns dispositivos, como por exemplo, tipos específicos de PDR (*Personal Digital Recorder*), podendo estar restrito a um tempo definido ou específico. Existindo relação com o provedor de serviço, os direitos podem ser seguramente transmitidos de uma parte a outra em ambiente TVA.

A TV-Anytime apresenta alguns cenários que podem ser utilizados para fins educacionais, tais como:

- Usuário que capta via *Wireless* um conteúdo de seu interesse e quer armazenar para si, ou compartilhar, posteriormente, com outras pessoas, independente do mesmo ser em forma de texto, imagem, vídeo, áudio, etc. Os conteúdos podem ser capturados, armazenados, distribuídos, redistribuídos e até modificados (desde que haja permissão) através de uma vasta gama de dispositivos PDR, PVR (*Personal Video Recorder*) e PDA (*Personal Digital Assistant*);
- Captura de um pacote interativo completo contendo aplicações, arquivos de áudio e vídeo separados, além de dados com diversos *links* de *websites* atualizados, sendo que o aplicativo em questão permite ao telespectador ir de um tópico a outro, revisar itens, classificá-los e movê-los para cima ou para baixo conforme níveis de competência, utilizando o mecanismo de armazenamento em PDR para reter a atenção dos telespectadores e para avaliar sua progressão através do conteúdo;
- Iniciar aquisição de conteúdo gratuitamente via emissora de TV de algum programa com aplicações interativas, e caso haja interesse em continuar o estudo, informações mais detalhadas e específicas sobre o assunto são disponibilizadas via pagamento;
- Uma aplicação de objeto de aprendizagem capturada em PDR que apresenta várias etapas. Alguns PDRs encontram-se conectados e pode haver a conexão com outros PDRs a fim de formar uma rede para troca e consulta de informações;
- Para alunos e ou pessoas com necessidades especiais, por exemplo, deficiência auditiva, tem-se opção de sincronizar mensagens textuais com imagens, em vez do áudio;
- Captura de informações e conteúdo em PDR para retransmissão, independente de ser na mesma sala, escola, cidade ou país e desde que se tenha autorização para distribuição.



### 3.7.1 Arquitetura do TV-Anytime em um Sistema de *Broadcast*

O *Personal Digital Recorder* (PDR) é o centro das especificações TV-Anytime e, portanto, um pré-requisito para essa especificação. Ele é o dispositivo do consumidor que inclui disco de armazenamento de alta capacidade. Com isso, o consumidor pode gravar conteúdos e assistir os mesmos independentemente da agenda da emissora (TVA, 2001). Porém, pode fazer uso ainda do *Network Digital Recorder* (NDR), que são dispositivos de armazenamento remoto, que podem incluir discos virtuais de armazenamento, localizados em servidores fora de casa (TVA, 2001).

Um sistema de *broadcast* que utilize o TVA pode ser visto como se tivesse três grandes elementos: um provedor de serviço que irá criar o conteúdo, um provedor de transporte que irá localizar e carregar o serviço e um dispositivo na parte do cliente que irá receber o serviço e armazenar os objetos para que possam ser visualizados pelo usuário posteriormente. A Figura 19 ilustra o caso onde o TVA é utilizado para um cenário *broadcast*.

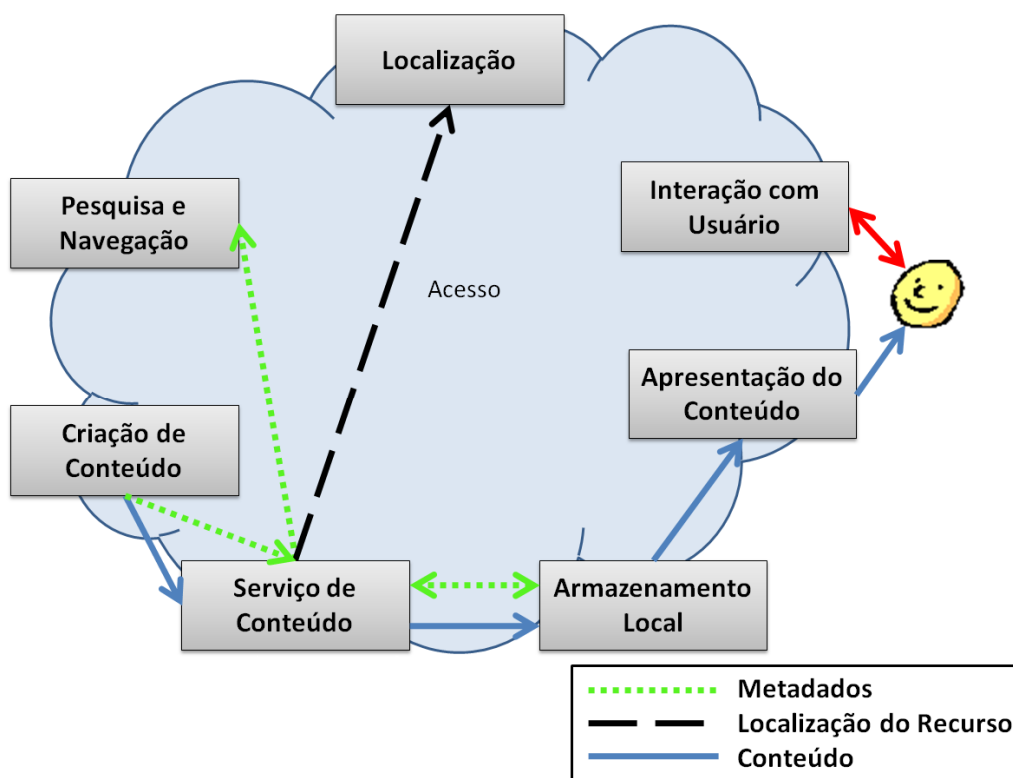


Figura 19. Sistema de *broadcast*. Adaptado de (TVA, 2003-b).

Cada caixa representa uma função do sistema e pode ser implementada em diferentes modos e por diferentes provedores de serviço. Implementações diferentes poderão levar a um ordenamento diferente das funcionalidades. As flechas da figura significam os fluxos de informações entre as funções do sistema.

A caixa Criação de Conteúdo é função típica de um estúdio de vídeos ou companhia de entretenimento. A caixa de fornecimento de Serviço de Conteúdo corresponde ao acesso e distribuição e também pela adição de metadados ao objeto. Normalmente essa tarefa é responsabilidade de uma emissora de televisão. Essas funções recém citadas são externas ao PDR.

As outras caixas representam funções que devem ser implementadas pelo PDR. As caixas Pesquisa e Navegação, Localização e Interação com Usuário são acionadas quando o usuário faz pesquisas por algum conteúdo. Armazenamento Local representa a capacidade do PDR de armazenar os objetos encontrados na etapa de busca. Por fim, a caixa Apresentação do Conteúdo tem a função de exibir o conteúdo dos objetos ao usuário, seja ele obtido a partir de armazenamento local ou pelo *broadcast* momentâneo.

A especificação TVA prevê três modelos funcionais ou perfis de dispositivo de usuário (TVA, 2001):

- Modelo 1: *Broadcast Model* (via entrega unidirecional, não tem canal de interatividade);
- Modelo 2: Canal de Retorno Intermitente (limitado, canal de interatividade ocasional);
- Modelo 3: *Bi-direcional Broadband* ('*always on*' network connectivity).

### 3.7.2 Modelagem de Metadados do TV-Anytime

A modelagem de metadados descrita nesta subseção é definida nas normas do *TV-Anytime Fórum*, que descrevem a arquitetura do TV-Anytime (TVA, 2003-a) e referentes à definição de seus metadados (TVA, 2003-b).

No ambiente TV-Anytime, a parte mais visível dos metadados são os descritores ou *hiperlinks* utilizados nos guias eletrônicos de programação ou em páginas Web. Essa é a informação que o consumidor ou um agente irá utilizar para decidir se irá ou não adquirir um determinado conteúdo (TVA, 2001). Ele define também um formato padrão para descrever os perfis de usuários, incluindo preferências de busca para facilitar filtros automáticos e aquisição de conteúdo por agentes no lado do consumidor (TVA, 2001).

A organização dos metadados do TVA foi planejada visando a separação entre conteúdo e fluxo de metadados. A integração é feita de modo transparente e pode ser feita tanto para as etapas de pós-produção, de entrega de conteúdo ou de consumo. Na etapa de produção de conteúdo, a atribuição de metadados ao conteúdo é feita a partir de uma coleta originada de diferentes fontes, eles são então agregados e editados em uma forma destinada ao usuário, é nessa etapa que são inseridos os metadados descritivos. A entrega é a fase que recebe metadados sobre a programação do conteúdo, assim como mecanismos de busca e navegação, é a etapa responsável por permitir a obtenção do conteúdo pelo usuário. A terceira etapa é referente ao consumo do conteúdo e metadados, nessa etapa são capturadas informações sobre as preferências do usuário, além de indicar os modos de armazenamento do conteúdo.

Os metadados do TVA foram divididos em quatro grandes categorias: conteúdo, instância, consumo e segmentação. A Figura 20 ilustra as principais tabelas de cada grupo e a Tabela 3.1 cita alguns exemplos de metadados contidos nessas categorias.

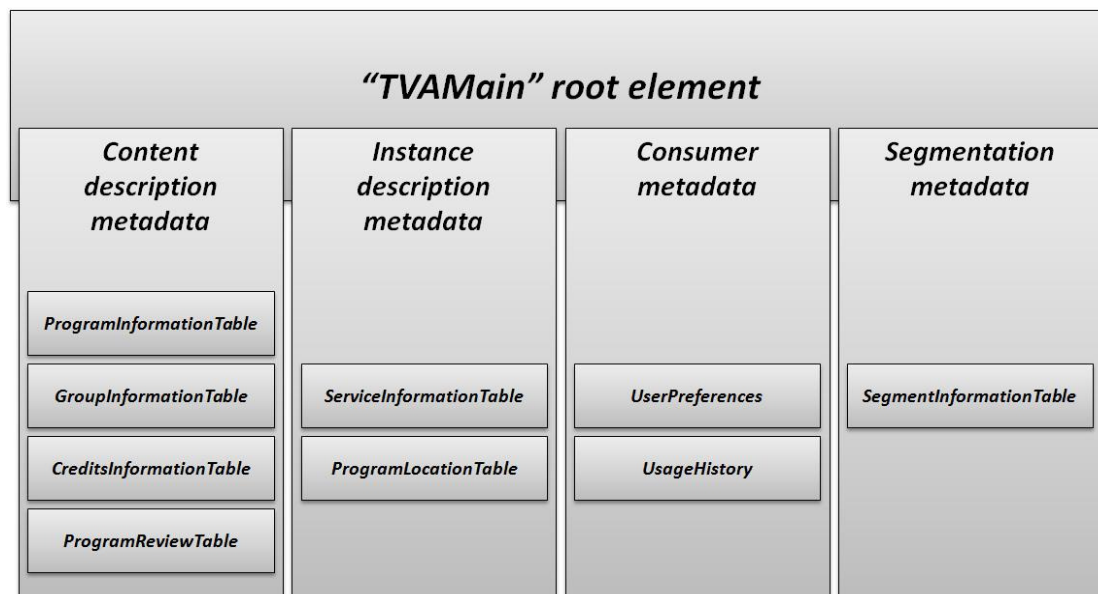


Figura 20. Documentos TV-Anytime com “TVAMain” como elemento raiz. Adaptado de (TVA, 2001).

Tabela 3.1: Categorias de Metadados do TV-Anytime.

<b>Tipo</b>	<b>Exemplo</b>
<b>Metadados de Descrição de Conteúdo</b>	
Básico Descritivo Informação de áudio/video  Informações de programa Informações de grupo Descritores de revisão de mídia Definições de metadados adicionais	CRIDType, TVAIDType Título, sinopse, gênero Formato de arquivo, Atributos de A/V, <i>bit-rate</i>  Tipo de variação Episódios de uma série Classificação, crítica sobre um filme Metadados que devem ser obrigatórios
<b>Metadados de Descrição de Instância</b>	
Entidades de localização de programa Localização de programa Informações de service	Programação (EPG) Uma entrada em um EPG Canal de TV, Canal 1 - Ciência
<b>Metadados de Usuário</b>	
Histórico de uso e preferências de usuário	Dados coletados sobre o usuário explícita ou implicitamente (e.g. usuário assiste notícias todos os dias às 21:00)
<b>Metadados de Segmentação</b>	
Descrição básica de segment Informações de segment Informações de grupo de segmentos Tabela de informações de segmentos	Título do segmento, sinopse do segmento Localização temporal do segmento Compilação de segmentos Tabela com todos os metadados de descrição de segmentos

Fonte: TVA, 2003-a e TVA, 2003-b.

Detalhando esses grupos, os metadados de descrição de conteúdo (*Content Description Metadata*) são uma categoria destinada à descrição do conteúdo e também

de suas variações. Essas variações podem ser, por exemplo, uma agregação de diferentes objetos (e.g. os capítulos de um curso). Os metadados descritivos são úteis para busca, navegação e seleção e para a personalização de conteúdo (combinados com as preferências do usuário).

O TVA define como requisitos de descrição de conteúdo um modelo no qual seja possível representar os seguintes conceitos:

- Um programa simples;
- Um programa com diferentes versões (por exemplo, edição completa, edições com cortes, etc.);
- Um programa que foi dividido em um número de partes para publicação (por exemplo, um filme com 3 horas exibido em duas partes em dias diferentes);
- Um programa que é a concatenação de uma sequência de outros programas identificados como um programa unificado;
- Uma série de programas que podem ser ordenados (como episódios de uma série em uma sequência numérica) ou não ordenados;
- Uma coleção de séries ou programas individuais;
- Uma publicação de um programa que pode ter atributos dependentes da publicação (por exemplo, um filme que exibe uma homenagem a um ator falecido recentemente e que teria uma descrição diferente).

Esta categoria é ainda dividida em quatro áreas, representadas por tabelas. Cada uma dessas tabelas pode conter metadados sobre mais de um objeto e, da mesma forma, um objeto pode ser descrito por mais de uma tabela.

A seguir estão listadas as tabelas que compõem os metadados de descrição de conteúdo (TVA, 2001):

- *ProgramInformationTable*: Essa tabela contém as informações mais básicas do conteúdo, como título, sinopse, palavras chave, etc., informações sobre a mídia, como formato de arquivo, *bit rate*, etc., e também informações sobre sua relação com outros objetos;
- *GroupInformationTable*: Essa tabela serve para indicar relacionamento entre conteúdos que formam um programa como um todo. Pode-se citar como exemplo, uma minissérie composta por vários capítulos. Cada capítulo será descrito pelos metadados da *ProgramInformationTable* e ao mesmo tempo a minissérie como um todo será descrita pelos metadados do *GroupInformationTable*. Da mesma forma, um conjunto de objetos de aprendizagem podem formar um curso, que seria descrito pelo *GroupInformationTable*;
- *CreditsInformationTable*: Armazena um mapeamento dos participantes na produção do objeto para identificadores únicos. Esses identificadores podem ser utilizados para facilitar as buscas;
- *ProgramReviewTable*: Essa tabela contém informações sobre as revisões e críticas realizadas sobre o objeto.

Em relação aos metadados de descrição de instância (*Instance Description Metadata*), eles são utilizados para ajudar na obtenção e consumo dos objetos e metadados pelo usuário. Guias de programação (e.g. EPG) e informações sobre a localização dos objetos são típicos metadados de instância.

A seguir estão listadas as tabelas que compõem os metadados de instância:

- *ProgramLocationTable*: Essa tabela é responsável por manter informações sobre como e quando os objetos podem ser obtidos. Uma de suas principais funcionalidades é a utilização de seus dados para criação de EPGs;
- *ServiceInformationTable*: Contém informações sobre os provedores do serviço que disponibilizam os objetos.

Já os metadados de usuário (*Consumer Metadata*) possuem estruturas para definir identificação de usuários, identificação de grupos de usuários, perfis de usuários (*profiles*) e histórico. Com esses dados é possível planejar conteúdo personalizado para cada tipo usuários, além de possibilitar a busca automática desse conteúdo baseada no histórico e preferências do usuário.

Essa categoria também possibilita o monitoramento de diversos tipos de ações tomadas pelo usuário, tais como pesquisa, filtragem de conteúdo, seleção de conteúdo, navegação e visualização.

Porém, prevê o respeito a privacidade do usuário. Segundo (TVA, 2001), a informação de perfil pessoal é somente propriedade do usuário final. A divulgação das informações privadas com outras partes sempre respeitará a disposição do usuário para tal.

Os metadados de segmentação (*Segmentation Metadata*) são utilizados para reestruturar o conteúdo para gerar novos modos de consumo e navegação para o objeto original. Permite a criação de segmentos e de grupos de segmentos a partir de um ou mais conteúdos de áudio e/ou vídeo. Pode-se também adicionar metadados descritivos para cada grupo de segmentos ou para segmentos individuais. A função desses metadados pode ser vista como o inverso da função dos metadados contidos na tabela *GroupInformationTable*, pois ao invés de agrupamentos eles dividem os objetos, possibilitando pesquisa por partes ou eventos dentro do mesmo.

Uma utilização para essa categoria poderia ser um resumo do conteúdo com algumas cenas enfatizando o que poderá ser visto no conteúdo principal. Esses metadados também podem ser utilizados para melhorar a indexação de conteúdo para busca (por exemplo, pode-se citar um vídeo de um curso, onde interessa a alguns alunos apenas os segmentos dos vídeos onde algumas palavras chave são pronunciadas, representando uma parte específica do assunto).

Estes metadados podem ser classificados em três tipos (TVA, 2001):

- Segmentação (*segmentation*): é definida como a divisão do conteúdo em diferentes partes (por exemplo, cenas). Os segmentos podem ser seções individualmente capturadas ou índices simples em um conteúdo longo;
- Destaques (*highlights*): são segmentos de conteúdo reproduzidos em uma sequência lógica, como, por exemplo, todos os gols em uma partida de futebol;
- Marcadores (*Book-marking* ou indexação pessoal): são índices gerados pelo usuário, colocados ao longo do conteúdo.

A única tabela que faz parte dessa categoria é a *SegmentInformationTable* e é a responsável por armazenar todas informações sobre a segmentação.

### 3.8 MXF (*Material eXchange Format*)

O MXF, padronizado pela SMPTE (*Society of Motion Picture and Television Engineers*) (SMPTE, 2010), é um formato aberto de arquivo, que objetiva o armazenamento e troca de conteúdo audiovisual, possuindo suporte a *timecode* e possibilita também a inclusão de metadados que descrevem o conteúdo encapsulado.

Suporta uma grande variedade de formatos de áudio e vídeo (chamado pelo padrão de “essência”), os quais ele encapsula. O padrão MXF não modifica e nem compacta o conteúdo, mas apenas envelopa (*wrap*), mantendo o conteúdo em um formato que pode ser reproduzido (WILKINSON e DEVLIN, 2002). Ao final do processo de encapsulamento desse conteúdo com dados de meta-informação, um novo arquivo, com extensão “*mxf*”, é gerado. Este padrão é suportado por vários fabricantes de equipamentos, como Sony e Panasonic. Como o MXF encapsula conteúdos de diferentes formatos e tipos de compressão, ele simplifica a integração e troca de dados entre sistemas, assim como a catalogação, por meio dos metadados existentes.

Sua principal utilização é nas emissoras de televisão, para armazenamento e catalogação do conteúdo digital em substituição as fitas cassetes. Entre as tarefas que o MXF possibilita estão:

- Possibilitar catalogação e armazenamento de trabalhos já concluídos com sua meta-informação, substituindo assim o uso de fitas cassetes;
- Armazenar arquivos num formato transferível (*streamable*), de forma que uma parte do arquivo já seja visível e editável enquanto o restante do arquivo está sendo transferido;
- Encapsular uma lista de arquivos a executar e guardar a sua informação de sincronização;
- Encapsular qualquer tipo de formato de compressão.

Conforme ilustrado na Figura 21, o arquivo MXF segue a seguinte estrutura (WILKINSON e DEVLIN, 2002):

- Um cabeçalho (*Header*), que inicia o arquivo e fornece as informações sobre o arquivo como um todo. Ao menos a primeira parte do arquivo deverá estar consistente para todas as implementações;
- Um corpo (*Body*), que contém os componentes essenciais do contêiner (conteúdos audiovisuais). Nos arquivos que tiverem mais de um componente essencial (por exemplo, um áudio e um vídeo), intercalação é fornecida pelo contêiner para que funcionalidade de *stream*, como a recuperação a partir de transferências incompletas, cortes de edição, etc., sejam possíveis;
- Um rodapé (*Footer*), que indica o fim do arquivo.

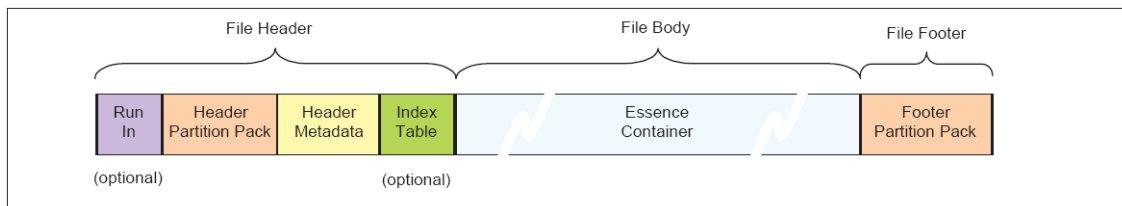


Figura 21. Componentes principais de um arquivo MXF simples (WILKINSON, DEVLIN, 2002).

Portanto, para exibir o conteúdo de um arquivo “mxh”, o player primeiramente lê o cabeçalho do arquivo para identificar o formato do conteúdo. O conteúdo é, então, processado pelo seu *codec* específico (por exemplo, um determinado tipo de vídeo é lido pelo seu *codec* daquele tipo).

O cabeçalho do arquivo MXF contém os seguintes componentes básicos (WILKINSON, DEVLIN, 2002):

- *Run-in*: necessário apenas em aplicações específicas onde seja necessário reconhecer a sequência de execução dos pacotes antes de iniciar a leitura do cabeçalho do arquivo;
- *Header Partition Pack*: fornece alguns metadados principais a sobre a estrutura do arquivo e também uma indexação simples do cabeçalho para permitir o acesso rápido do *decoder*;
- *Header Metadata*: esse possui duas partes: (1) *Structural Metadata*, uma estrutura de objetos compatível com as classes AAF (*Advanced Authoring Format*) e (2) *Descriptive Metadata*, definido como o *plugin* para permitir metadados adicionais para descrição do arquivo.
- *Index Table*: permite o acesso rápido a qualquer componente do container de conteúdo através de um valor de *offset*, indicando o posicionamento daquele conteúdo. Embora opcional, é altamente recomendado para aumentar o desempenho.

O corpo pode incluir um único ou intercalar vários componentes essenciais, ou seja, o conteúdo em si. Cada contêiner é escrito como um *plugin* para possibilitar a inclusão de qualquer formato padronizado de container. Para um container ser aceito como *plugin* MXF, algumas restrições são definidas para assegurar que o padrão seja respeitado (WILKINSON, DEVLIN, 2002):

- Deve encapsular os componentes essenciais com a codificação KLV (*Key, Length, Value*) (SMPTE, 2001), utilizando chaves registradas publicamente;
- Deve prover intercalação dos componentes sobre uma duração definida;
- A especificação deve ser disponibilizada publicamente por uma entidade reconhecida internacionalmente.

O rodapé deve conter ao menos o pacote de partição de rodapé (*Footer Partition Pack*), indicando fim do arquivo. É opcional também incluir uma repetição dos Metadados do Cabeçalho (*Header Metadata*) e da tabela de índices (*Index Table*).

### 3.8.1 Metadados do Arquivo MXF

Em relação aos metadados, podem ser referentes a quatro áreas, conforme ilustrado na Figura 22.

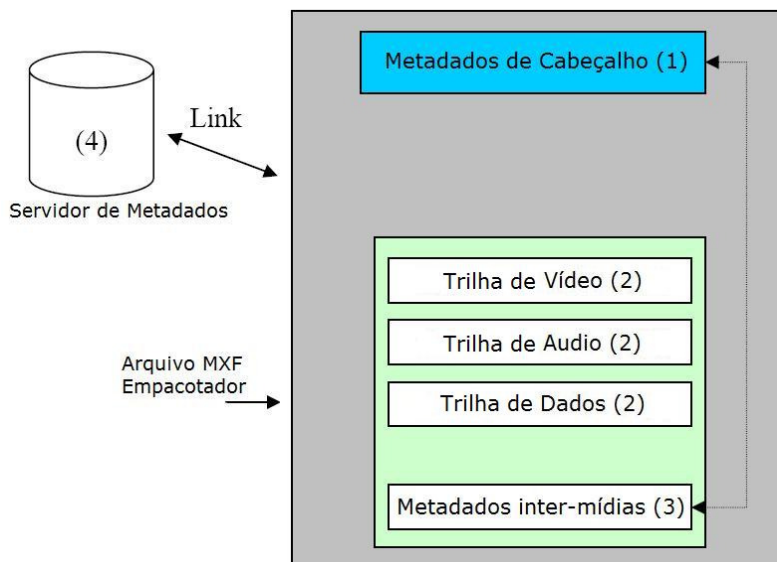


Figura 22. Áreas de metadados em MXF (adaptado de SMPTE EG apud KALLIOJA, 2006).

A área de metadados do cabeçalho (1) contém metadados que se referem ao arquivo MXF como um todo. Os metadados embutidos em cada mídia (2) se referem aos metadados daquele conteúdo específico, enquanto os metadados intermídias (3) se referem ao relacionamento entre as mídias encapsuladas. Por fim, um banco de dados externo, poderá conter metadados e indexar os arquivos MXF (4).

Os metadados do MXF são ainda categorizados em estruturais e descritivos. Os metadados estruturais descrevem a sincronização das diferentes trilhas e propriedades do conteúdo como taxa de amostragem, tamanho das imagens e organização dos canais de áudio (DEVLIN e WILKINSON, 2004 apud KALLIOJA, 2006). Os metadados descritivos se referem ao conteúdo da informação, sua catalogação e anotações do usuário. Esses metadados formam o padrão SMPTE 380M, *Descriptive Metadata Scheme-1* (DMS-1). A Tabela 3.2 lista os conjuntos de metadados descritivos desse padrão.

Tabela 3.2. Conjuntos de metadados DMS-1 e seu uso.

<b>Grupo</b>	<b>Descrição</b>
<i>Titles</i>	Fornece diferentes classes de títulos para o conteúdo audiovisual
<i>Identification</i>	Fornece uma identificação para produção
<i>Group Relationship</i>	Utilizado para agrupar conteúdos, como episódios de uma série
<i>Branding</i>	Define a marca a que o conteúdo de áudio e vídeo pertence
<i>Event</i>	Define eventos incluindo a data e hora inicial e final
<i>Award</i>	Fornece indicações históricas de prêmios fornecidos ao material produzido
<i>Caption Description</i>	Descrições como closed-caption e legendas



<i>Annotation</i>	Fornece informações adicionais, como sinopse ou palavras chave
<i>Setting Period</i>	Descreve o período em que a cena é localizada, por data/hora ou palavra chave
<i>Scripting</i>	Contém roteiros associados ao clip
<i>Classification</i>	Identifica o esquema utilizado para anotação
<i>Shot</i>	Fornece uma descrição em texto simples da cena
<i>Key Point</i>	Diferencia pontos chave como palavras chave, sons, fotogramas
<i>Participant</i>	Atribui o estado do participante como pessoas, grupos ou organizações
<i>Person</i>	Identifica uma pessoa
<i>Organization</i>	Identifica uma organização
<i>Location</i>	Descreve locais como local de câmera ou local de tomada de cena
<i>Address</i>	Endereços de pessoas, organizações ou locais
<i>Communications</i>	Identifica informações de contato e endereço de pessoas, organizações e locais
<i>Contract</i>	Fornece informações mínimas necessárias para identificar qualquer informação contratual
<i>Rights</i>	Inclui informação de direitos autorais
<i>Picture Format</i>	Identifica resolução e formato de exibição para a produção como um todo
<i>Device Parameters</i>	Identifica dispositivos utilizados para capturar ou criar conteúdo de A/V em um videoclipe
<i>Name-Value</i>	Fornece listas como pares nome-valor para itens registrados no dicionário SMPTE
<i>Processing</i>	Identifica o número e tipo de processamentos que o clipe pode ter sido submetido
<i>Project</i>	Fornece os dados equivalentes aos do <i>clapperboard</i>
<i>Contacts List</i>	Consiste nos grupos de pessoas, locais e organizações
<i>Cue Words</i>	Descreve informação verbal ou textual usado pelo time de produção para sugerir corretamente o programa ou um item do programa.

Fonte: Adaptado de SMPTE 380M apud KALLIOJA, 2006.

A Tabela 3.3 detalha as propriedades de grupo *Titles*.

Tabela 3.3. Metadados de Propriedades do Grupo *Titles*.

<b>Propriedade</b>	<b>Definição</b>
<i>Extended Text Language Code</i>	O código longo que representa a linguagem e opcionalmente a variância utilizada para o texto
<i>Thesaurus Name</i>	O nome de um vocabulário especializado das palavras selecionadas ou conceitos para um campo em particular, por exemplo, um catalogador particular, indexador ou dicionário de sinônimos
<i>Main Title</i>	Título principal da produção
<i>Secondary Title</i>	Segundo título da produção
<i>Working Title</i>	Um título, possivelmente temporário, da produção
<i>Original Title</i>	O título original da produção
<i>Version Title</i>	Versão do título da produção

Fonte: Adaptado de SMPTE apud KALLIOJA, 2006.

## 4 PROPOSTAS PARA INTEROPERABILIDADE

O objetivo deste capítulo é apresentar as principais contribuições desta dissertação. Inicia com as recomendações para criação de objetos de aprendizagem interoperáveis para as plataformas Web, televisão digital e dispositivos móveis (Seção 4.1). Segue com a descrição do *Interop*, um *framework* desenvolvido para facilitar a criação de objetos de aprendizagem que seguem as recomendações citadas (Seção 4.2). Por fim, apresenta os dois conjuntos de metadados propostos como extensão aos padrões educacionais, tendo como ganho a possibilidade de catalogação de objetos de aprendizagem desenvolvidos para diferentes plataformas, assim como a segmentação dos mesmos (Seção 4.3).

### 4.1 Criação de Objetos de Aprendizagem Interoperáveis

No Capítulo 2 (Padrões Inerentes às Plataformas) foram apresentados os recursos disponíveis em cada uma das plataformas estudadas neste trabalho (TV digital, Web e dispositivos móveis). Como resultado, identificou-se formatos de mídia e linguagens em comum nas três plataformas e que podem ser utilizados para a reutilização do conteúdo entre elas.

A Tabela 4.1 apresenta, de forma simplificada, um quadro comparativo dos recursos existentes em cada uma das plataformas.

Tabela 4.1. Quadro comparativo dos recursos de cada plataforma.

	WEB	TV Digital	Móveis
<b>Imagens</b>	JPEG, PNG, BMP, SVG, GIF, etc.	JPEG, PNG	JPEG, PNG, BMP, GIF
<b>Áudio</b>	AAC, MP3, MIDI, WAV, MP4 e outros	AAC, MP4, WAVE, AIFF	AAC, AAC+, AMR-NB, AMR-WB
<b>Vídeo</b>	Diversos formatos, utilizando <i>plugins</i>	H.264 em formato "ts"	<i>3GP, MPEG-4 Visual, Real Video</i>
<b>XHTML</b>	Sim	Sim	Sim
<b>CSS</b>	Sim	Sim (Perfil Simplificado)	Sim ( <i>Mobile Profile</i> )
<b>Linguagem Script</b>	<i>ECMAScript</i>	<i>LUA Script</i> (Obrigatório) <i>ECMA Script</i> (Opcional)	<i>ECMAScript</i> ( <i>Mobile profile</i> )
<b>NCL</b>	Não	Sim	Não

<b>Java</b>	<i>Applets</i>	<i>JavaDTV</i> (Ainda não disponível)	<i>Midlets</i>
<b>Flash</b>	Sim	Não	Sim ( <i>Flash Lite</i> )
<b>Pdf</b>	Sim	Não	Sim

As próximas subseções apresentam um comparativo das principais recomendações de usabilidade entre os dispositivos, assim como as recomendações para os diversos níveis de padronização possíveis para a interoperabilidade entre as três plataformas.

#### 4.1.1 Interface com o usuário

Interfaces para dispositivos móveis, Web e de TV digital possuem diferentes recomendações de usabilidade, que refletem na construção da interface. Enquanto na Web e no celular, são sugeridas cores claras de fundo e cores escuras de fonte, na TV digital o inverso é recomendado (cor escura de fundo e cor de fonte clara). Em relação ao padrão de fonte em si, cada *browser* de celular vem com sua própria definição de fontes. Para a TV digital são recomendadas *Gill Sans*, *Tiresias* e *Frutiger*, por serem fontes não serifadas e, portanto, mais legíveis (BBC, 2006). Já na Web, várias fontes podem ser utilizadas. Em relação ao tamanho da fonte, na TV digital são sugeridos no mínimo 18 pontos para conteúdo texto (BBC, 2006), devido à resolução, e no celular 12 *pixels* devido ao tamanho de tela. Para a Web, no entanto, não foram encontradas recomendações nesse sentido. A Tabela 4.2 sumariza as principais recomendações de usabilidade.

Tabela 4.2. Principais recomendações de usabilidade.

<b>Dispositivo</b>	<b>Web</b>	<b>TV Digital</b>	<b>Móveis</b>
Cor da Fonte	Cores escuras	Cores claras	Cores escuras
Cor do Fundo	Cores claras	Cores escuras	Cores claras
Fonte recomendada	<i>Arial</i> , <i>Times</i> , <i>Verdana</i> , <i>Georgia</i> , outras	<i>Gill Sans</i> , <i>Tiresias</i> e <i>Frutiger</i>	Família Sans-serif
Tamanho de Fonte	Não encontradas recomendações específicas	Título: 36 pt Menus: 20 pt Texto: 22 pt Botões: 18 pt	Título 1: 18 px Título 2: 16 px Texto: 12 px

Observa-se, portanto, que embora o conteúdo apresentado nos diferentes dispositivos seja o mesmo, recomenda-se que a formatação e apresentação desse conteúdo se dêem de forma diferenciada, utilizando as características visuais mais adequadas à plataforma onde o objeto de aprendizagem será exibido.

#### 4.1.2 Recomendações para Conteúdo Texto

Entre os conteúdos de tipo texto, a proposta deste trabalho é a utilização da linguagem XHTML, juntamente com folhas de estilos CSS e linguagens *Script*.

Conforme descrito anteriormente, além de não consumir muito espaço, o uso do XHTML para exibição de texto formatado é padronizado para a Web, TV digital e dispositivos móveis. Na TV digital, o XHTML é inclusive recomendado para interoperabilidade com os principais sistemas de TV digital existentes no mundo.

Além disso, embora não seja obrigatório pelas normas do SBTVD, é provável que os *set-top boxes* sejam disponibilizados com navegadores Web embutidos. Uma vez que isso ocorra, os objetos de aprendizagem desenvolvidos poderão ser acessados também por esses *browsers* sem necessidade de novas modificações.

Já o *Cascading Style Sheets* (CSS) é bastante recomendado para possibilitar a personalização da interface em cada ambiente conforme as recomendações de usabilidade específicas de cada plataforma. Dessa forma, o servidor de aplicação retorna um arquivo de estilo diferente a partir da identificação do tipo de dispositivo que está requisitando a página Web. O *web designer* define um arquivo de estilos para cada dispositivo somente uma vez e pode reutilizá-lo em todas as páginas posteriormente criadas que utilizam aquele leiaute. A identificação do tipo de dispositivo e da adaptação da folha de estilos pode ser realizada no lado servidor, por *frameworks* como o *Interop*, que será descrito posteriormente.

Uma vez que tanto o XHTML como o CSS representam conteúdos estáticos, limitando a possibilidade de criação de lógica dinâmica, matemática e procedural no lado do cliente, podem ser utilizadas linguagens *scripts* para superar essas limitações. Conforme visto, o padrão ECMAScript estende as funcionalidades do XHTML na Web, celulares e TV digital, porém, com diferentes versões. Na Web, o padrão mais comum é o ECMA-262, enquanto que dispositivos móveis e TV digital suportam subconjuntos desse padrão. Porém, como a presença do ECMAScript nos *set-top boxes* é opcional (ABNT, 2008-f), pode ser utilizada a linguagem *Lua Script* em seu lugar, com a desvantagem de ter o mesmo algoritmo programado em duas linguagens diferentes (duas versões).

### 4.1.3 Imagens

Quanto às imagens, para a TV digital os formatos PNG, JPEG e H.264/MPEG-4 AVC “*I-Picture*” são de suporte obrigatório em todos os receptores. Nos dispositivos móveis, os formatos GIF, JPEG e PNG são suportados pela grande maioria dos dispositivos móveis atuais, conforme pode ser observado nas configurações dos celulares no site de seus fabricantes. Adicionalmente, o uso dos formatos GIF 89a e JPEG em conteúdos para Web móvel é uma recomendação W3C.

Com base nisso, para interoperabilidade de imagens, recomenda-se a utilização do formato JPEG, por ser padrão nos três ambientes. O formato PNG também pode ser utilizado. Embora não padronizado para dispositivos móveis, é amplamente suportado nos dispositivos móveis existentes no mercado e é padrão para TV digital e Web.

### 4.1.4 Vídeo

O padrão de compressão de vídeo definido para o SBTVD foi o H.264/AVC. Nos dispositivos móveis, o formato *3gp* com codificação H.263 é ainda o mais utilizado e, por isso, aqui recomendado. O formato de vídeo *mp4* é um *codec* de vídeo opcional em vários padrões recentes, no entanto, é amplamente suportado nos mais recentes dispositivos móveis disponíveis no mercado, juntamente com o *3gp*.

Para as implementações efetuadas neste trabalho, os vídeos foram armazenados em H.264 para TV digital e para Web, enquanto que para execução nos dispositivos móveis os mesmos foram convertidos para o formato *3gp* com compressão H.263, uma vez que ainda é o formato de vídeo mais difundido nesse ambiente.

#### 4.1.5 Áudio

Em relação ao áudio, as normas SBTVD definem como suporte obrigatório o formato MPEG-4 Áudio AAC-LC, e como opcionais os formatos MPEG-2 Áudio AAC LC/BC, PCM (AIFF-C), MPEG-1 *audio layer 3* (MP3). Para dispositivos móveis, são frequentemente utilizados diversos *codecs* de áudio, como AMR-NB, AMR-WB, *RealAudio Voice*, *RealAudio 7*, 8 e 10, MP3, AAC (MPEG-4 *Advanced Audio Coding*), AAC+ e eAAC+.

Por ser comum aos três ambientes, optou-se pela utilização do formato de áudio AAC.

#### 4.1.6 Outros formatos de mídia

Os formatos de mídia apresentados até o momento possuem suporte para as três plataformas consideradas neste trabalho. Já outros formatos bastante difundidos na Web, como o *Flash* e *PDF*, possuem *players* mais restritivos em termos de recursos em grande parte dos dispositivos móveis (nas versões chamadas “*lite*”) e ainda não são contemplados para a TV digital.

Dessa forma, documentos em formato PDF precisariam ser convertidos para XHTML para possibilitar sua interoperabilidade total, enquanto que animações em *flash* poderiam ser convertidas para vídeo e, quando dotadas de interatividade com usuário, complementadas por linguagens *script*. Porém ferramentas para automatizar esse tipo de trabalho não foram contempladas no escopo deste trabalho, ficando como sugestão de uma possível continuidade do mesmo.

A linguagem Java, apesar de sua adoção pelas três plataformas e, conseqüentemente, uma alternativa promissora para desenvolvimento interoperável, conforme já citado anteriormente, ainda não está disponível para TV digital, estando em fase de aprovação da normatização e em desenvolvimento pela indústria. Assim, ainda não há ambientes de simulação e testes para o Gingga-J. Dessa forma, a análise mais aprofundada da utilização do Java para interoperabilidade ficará como outra possível continuidade e ampliação deste trabalho.

## 4.2 Mecanismos de adaptabilidade

Na construção de objetos de aprendizagem interoperáveis, mesmo utilizando mídias comuns às três plataformas, algumas adaptações necessitam ser realizadas em função, por exemplo, dos padrões de usabilidade e tamanho de tela do dispositivo que está requisitando a página.

Embora a maioria das adaptações possa ser facilmente obtida através do uso de folhas de estilo (CSS), o mesmo é estático. Porém, existem algumas adaptações que necessitam ser realizadas em tempo de execução, pois dependem da plataforma que está requisitando a página. A inclusão do CSS adequado ao dispositivo que está realizando a requisição é um exemplo.

Além disso, muitas vezes é interessante inserir imagens, textos, áudios e vídeos diferenciados para cada plataforma, visando obter uma melhor apresentação ou desempenho no uso de recursos. Por exemplo, os dispositivos móveis e de TV digital não são capazes de reproduzir todos os tipos de arquivos existentes na Web. Portanto, poder-se-ia incluir algum recurso adicional na plataforma Web e suprimi-lo na plataforma de TV digital. Além disso, podem ser utilizadas imagens em tamanhos reduzidos nos dispositivos móveis, melhor adequando ao seu menor tamanho de tela ou taxa de transferência.

Uma vez que documentos HTML/XHTML, assim como as demais mídias recomendadas, não fornecem meios para adaptar sua apresentação em tempo de execução, optou-se por especificar e criar um *framework* para facilitar a criação de hiperdocumentos interoperáveis e que, ao mesmo tempo, permita maior flexibilidade na produção de conteúdo XHTML de forma diferenciada para cada plataforma quando necessário.

Esse *framework* foi denominado *Interop*, sendo que o mesmo foi criado nesta dissertação juntamente com o trabalho de conclusão de curso de VARELLA (2009-a). O *Interop* é composto dos seguintes módulos:

- **Módulo Servidor:** servidor de aplicação Web baseado em JavaEE (*Java Enterprise Edition*), responsável pela identificação automática do tipo de plataforma cliente e pelas adaptações necessárias em tempo de execução;
- **Módulo *Taglibs*:** Uma biblioteca de *tags* para facilitar a criação de objetos de aprendizagem interoperáveis Web;
- ***Website Downloader*:** extrai os objetos interoperáveis do servidor Web no formato para TV digital, salvando-os em formato XHTML em um diretório, possibilitando posterior envio via *broadcast* pelas emissoras ou até mesmo execução via *pen drive*. Esse recurso se fez necessário uma vez que a maioria dos *set-top boxes* atuais ainda não possuem a funcionalidade de acessar aplicativos diretamente pelo canal de interatividade;
- ***Website Converter*:** automatiza parte da conversão de objetos de aprendizagem desenvolvidos inicialmente para a Web, em formato HTML, para o formato interoperável, utilizando as bibliotecas de *tags* desenvolvidas.

Esses módulos são detalhados a seguir.

#### **4.2.1 Módulo Servidor**

Uma vez que o *Interop* é um *framework* para hiperdocumentos, se faz necessário um módulo servidor, o qual é responsável principalmente pelo processamento de páginas Web dinâmicas JSP (*Java Server Pages*), retornando o conteúdo adequado após analisar qual a plataforma que está requisitando o mesmo. A escolha da plataforma JavaEE juntamente com JSP partiu do fato de sua grande adoção para desenvolvimento Web, sendo que algumas vantagens que podem ser citadas são a portabilidade, facilidade de programação, flexibilidade, escalabilidade e eficiência, conforme detalhado em (VARELLA *et al.*, 2009-b).

Para identificação do tipo de cliente, o *Interop* utiliza as informações dos cabeçalhos contidos nas requisições HTTP<sup>13</sup>. Conforme Figura 23, a informação mais importante na identificação do dispositivo cliente é o cabeçalho chamado *user-agent*. Esse contém o nome e algumas informações adicionais sobre a aplicação do cliente que está realizando o acesso ao objeto (nesse caso, indicando o navegador *Mozilla Firefox 3.0.5*). Outra informação relevante é o *accept*, que indica quais os tipos de conteúdo em que a aplicação do cliente é capaz de executar.

Ao analisar esses cabeçalhos, algumas palavras chaves serão buscadas para identificação da plataforma cliente. Por exemplo, palavras como *windows ce*, *iphone*, *iemobile*, *symbian*, *mini*, *phone*, *pda* ou se iniciar por sequências como *cell*, *lg-*, *mobi*, *moto*, *noki*, entre outras, será uma indicação de que o dispositivo alvo é um dispositivo móvel.

```

host: localhost:16396
user-agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; pt-BR; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5
accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
accept-encoding: gzip,deflate
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
keep-alive: 300
connection: keep-alive
cookie: JSESSIONID=9309d37ec579609a1f3f64e82700
cache-control: max-age=0

```

Figura 23. Elementos do cabeçalho HTTP, de uma requisição do Firefox 3.0.5.

O gerenciamento de sessão do usuário também é realizado por esse módulo, onde a identificação do tipo de cliente é realizada na primeira requisição de página ou a cada vez que sua sessão expirar (por um período de inatividade, por exemplo).

Outra funcionalidade do servidor de aplicação é a adaptação de mídias. Como já foi citado, algumas vezes é desejável utilizar um tipo ou versão de mídia diferente para cada tipo de dispositivo alvo, devido à diferentes limitações ou recomendações de usabilidade. Um exemplo prático é o retorno de um arquivo CSS diferente para cada plataforma.

A solução adotada até a versão atual do *framework* baseia-se em acrescentar ao nome do recurso um sufixo indicando a plataforma. Esse recurso alternativo deverá estar no mesmo diretório do recurso que está escrito no XHTML, porém adotando os sufixos “*\_mobile*” e “*\_tvd*” para dispositivos móveis e TV digital, respectivamente. Por exemplo, para o arquivo padrão *figural.jpeg*, poderão ser disponibilizados os seguintes arquivos específicos: *figural\_mobile.jpeg* e *figural\_tvd.jpeg*. Dessa maneira, quando o recurso *figural.jpeg* for referenciado no XHTML, o servidor verificará se existe um arquivo com o mesmo nome, porém, com o sufixo “*\_mobile*” (no caso de ser um dispositivo móvel) ou “*\_tvd*” (se for uma requisição via TV digital). Se existir, essa imagem específica será então utilizada. Caso contrário, utilizar-se-á a imagem padrão.

#### 4.2.2 Módulo *TagLibs*

Na escolha de uma tecnologia para desenvolvimento Web, um dos fatores que deve ser considerado é a facilidade de uso da solução. Normalmente os *web designers*, que também terão que utilizar a tecnologia, não possuem habilidades com linguagens de programação. Logo, é interessante que a codificação seja a mais próxima possível do

<sup>13</sup> Disponível em: <<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>>. Acesso em: ago. 2010.

uso do XHTML, já que essa linguagem é de conhecimento comum também a esses desenvolvedores. Isso foi resolvido criando um módulo de *taglibs*.

*Taglib* é um conjunto de *tags* cuja finalidade é justamente abstrair do desenvolvedor de objetos de aprendizagem a programação procedural dinâmica (métodos e classes). A escolha por *taglibs* deveu-se a simplicidade e praticidade, pois sua utilização se dá da mesma forma que as *tags* HTML padrão e, dessa forma, *web designers* sem conhecimento de programação Java também podem utilizar essa biblioteca de *tags*.

A Figura 24 exemplifica o uso de uma *tag* genérica, chamada *tagName*, a qual pertence a uma *taglib* fictícia, representada pelo prefixo *taglibPrefix*. Os prefixos também são de uso comum na tecnologia de *taglibs* para facilitar o uso.

```
<taglibPrefix:tagName att1="valor" att2="valor">
  Texto que será processado.
</taglibPrefix:tagName>
```

Figura 24. Exemplo de uso de uma *taglib* fictícia

No caso do *framework Interop* foi adotado o prefixo “i” e foram previstas e criadas *tags* para os tipos de mídia elencados na Seção 4.1, como `<img>` e `<video>`, além de outras para flexibilizar o uso de *tags* HTML, como `<div>`, `<a>` e `<p>`. Essas *tags* foram pensadas a ter nomes semelhantes às do HTML e XHTML, o que facilita o uso por *web designers* (que já conhecem essas especificações), reduzindo a curva de aprendizado.

Outro recurso da *taglib Interop* é a geração condicional de código XHTML. Essa funcionalidade serve para as situações em que é desejável que um determinado conteúdo ou trecho do objeto de aprendizagem seja exibido somente em uma determinada plataforma. Esse recurso pode ser utilizado, por exemplo, para incluir conteúdo adicional e que funciona na plataforma Web, como *Flash* e *Pdf*, mas que não é suportado na TV digital ou móvel.

Para permitir essa funcionalidade, foram previstos três atributos na *taglib*: *displayMobile*, *displayTV* e *displayWeb*, os quais são do tipo lógico. Esses atributos podem ser utilizados em todas as *tags* e aceitam os valores *true* e *false*, sendo que o valor padrão é *true*. Ou seja, os atributos só precisam ser explicitados quando desejado inibir a geração de código em alguma das plataformas cliente.

A Tabela 4.3 apresenta as principais *tags* previstas pelo *Interop* e a forma de uso. Detalhes da implementação podem ser consultados em (VARELLA, 2009-a).

Tabela 4.3. Principais *tags* suportadas pelo *framework Interop*.

Tag a	<p>Criação de links entre páginas ou regiões de páginas.</p> <p>Equivalente a <i>tag</i> <code>&lt;a&gt;</code> do HTML e permite geração dinâmica das referências para cada plataforma.</p> <p>Uso: <code>&lt;i:a href="endereco.html"&gt;Texto do link&lt;/i:a&gt;</code></p>
Tag div	<p>Equivalente à <i>tag</i> <code>&lt;div&gt;</code> do HTML, que é responsável pela criação de regiões dentro dos documentos.</p> <p>Essas regiões são comumente utilizadas para compor o leiaute das páginas e, no caso do <i>Interop</i>, pode ser utilizada para o caso de necessidade em inibir a</p>



	<p>geração de código XHTML para alguma plataforma cliente em específico.</p> <p>Uso:</p> <pre>&lt;i:div displayMobile="true false" displayTVD="true false"&gt;</pre> <p style="text-align: center;"><i>Conteúdo da div</i></p> <pre>&lt;/i:div&gt;</pre>
<i>Tag img</i>	<p>Equivalente à tag &lt;img&gt; do HTML, que é responsável pela exibição de imagens dentro das páginas.</p> <p>Permite a geração dinâmica da referência às imagens e que alguma imagem possa ser exibida condicionalmente, dependendo da plataforma cliente.</p> <p>Uso: &lt;img src="figura.png"/&gt;</p>
<i>Tag link</i>	<p>Equivalente à tag &lt;link&gt; do HTML, que é responsável pela inserção de recursos externos.</p> <p>É utilizada para importar arquivos <i>scripts</i> e arquivos CSS diferentes para cada plataforma cliente.</p> <p>Uso: &lt;i:link rel="stylesheet" href="estilo.css" type="text/css"/&gt;</p>
<i>Tag p</i>	<p>Equivalente à tag &lt;p&gt; do HTML, que normalmente é utilizada para demarcar parágrafos. Seu uso é recomendado quando desejado que um trecho seja gerado ou não em função da plataforma cliente.</p> <p>Uso: &lt;i:p displayTVD="false"&gt;Texto do parágrafo&lt;/i:p&gt;</p>
<i>Tag vídeo</i>	<p>Não existe no HTML 4.0 / XHTML 1.1 uma <i>tag</i> destinada à execução de vídeos, logo essa é uma inovação embasada nas recomendações apresentadas cuja principal finalidade é simplificar a geração do código.</p> <p>Na Web o código necessário para se embutir um vídeo em uma página XHTML é extenso e complicado, além de que depende da inclusão de um arquivo externo com funções <i>JavaScript</i>.</p> <p>Para celulares, será apenas criado um <i>link</i> ao endereço do vídeo, assim ao acessar o endereço o aparelho irá utilizar o <i>player</i> padrão instalado para sua exibição.</p> <p>Já para a TV digital, o mesmo precisa ser referenciado por um documento NCL, que irá realizar a execução do vídeo.</p> <p>Uso:</p> <pre>&lt;i:video webHref="rtsp://host/movie.mp4" tvdHref="rtsp://host/movie.mp4" mobHref="rtsp://host/movie.3gp" width="240" height="320" controller="false" value="Link para o vídeo"/&gt;</pre>

Fonte: VARELLA, 2009-a.

A Figura 25 exemplifica o uso da tag <a>, criando um *link* e, nesse caso, exibindo apenas para os dispositivos móveis (exemplificando também o uso dos atributos de geração condicional de conteúdo). Cabe citar que todo o conteúdo contido na *tag* que contenha os atributos condicionais (inclusive outras *tags*) sempre serão afetadas. Essa

característica é possível graças à natureza hierárquica dada às páginas introduzida pelo uso das regras de formatação do XHTML.

```
<i:a href="Second.jsp" displayWeb="false" displayTv="false">  
    Este texto irá aparecer apenas nos dispositivos móveis.  
</i:a>
```

Figura 25. Exemplo de geração condicional de código XHTML.

### 4.2.3 Website Downloader

Visto que um dos objetivos é a possibilidade de acesso aos objetos de aprendizagem também a partir de receptores de TV digital, surge a necessidade de criar formas de acesso ao conteúdo também para dispositivos que não possuam canal de interatividade. O presente módulo foi planejado para permitir o envio dos objetos de aprendizagem pelo sinal de *broadcast* das emissoras.

Para isso, foi realizado o desenvolvimento de uma nova aplicação, chamada de *Website Downloader*, também desenvolvida em Java. Ela realiza requisições HTTP ao servidor e, a partir da página inicial (*index.jsp*), requisita recursivamente todas as páginas e recursos do *web site*. Após isso, irá gravar as páginas em XHTML e modificar as referências aos recursos e outras páginas para que se tornem relativas, tornando possível a navegação local, sem a ajuda de um servidor Web. Depois de terminado o *download*, uma emissora poderá enviar o objeto de aprendizagem via *broadcast* (ou permitindo até mesmo acessá-lo a partir de um *pen drive* conectado diretamente ao *set-top box*). Dessa forma, mesmo usuários cujo *set-top box* não tenha canal de interatividade, poderão visualizar os objeto de aprendizagem.

A Figura 26 ilustra o funcionamento da aplicação. Primeiramente, a página é requisitada diretamente do servidor Web (1), a qual será processada e armazenada, no formato de envio à TV, em um diretório local (2). O próximo passo (3) será efetuado pela antena de transmissão (pelos dispositivos que a controlam), a qual irá requisitar o diretório que foi salvo no passo (2), para que possa ser enviado por *broadcast* aos usuários (4). Pode-se ainda, salvar o objeto de aprendizagem em um *pen drive* (5) e acessá-lo diretamente pelo *set-top box*. É possível observar um círculo envolvendo o servidor, o diretório e a aplicação, sinalizando que todas essas entidades (sendo que a antena de transmissão também pode fazer parte desse ambiente) podem estar localizadas no mesmo servidor.

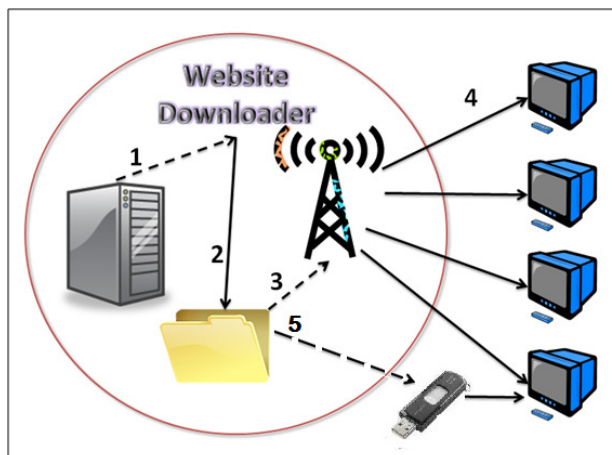


Figura 26. Funcionamento do *Website Downloader* (VARELLA, 2009-a).

A Figura 27 corresponde à tela principal da aplicação, nela é possível escolher entre informar a URL do site (para sites na *web*) ou o diretório do site (para sites locais). Além disso, pode-se escolher através de qual página o programa deve iniciar o *download* do site e em que diretório ele deverá ser salvo.

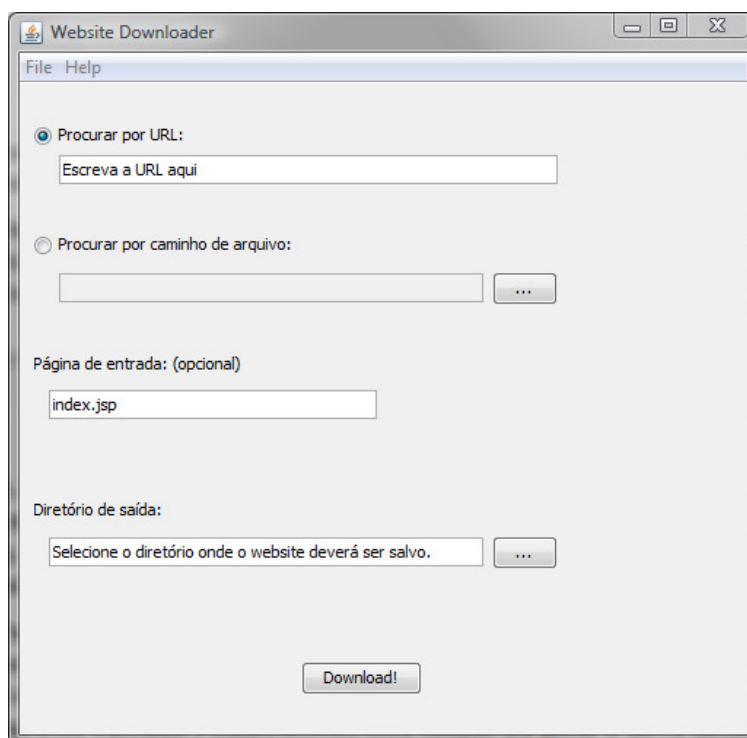


Figura 27. Tela inicial *Website Downloader* (VARELLA, 2009-a).

Neste momento, o *Website Downloader* está preparado para realizar o *download* apenas dos objetos de aprendizagem desenvolvidos para os estudos de caso apresentados no Capítulo 5, mas pode ser expandido por trabalhos futuros.

#### 4.2.4 *Website Converter*

Como foi visto, é necessário seguir algumas regras na construção das páginas Web para permitir o processamento das mesmas por parte da aplicação que irá hospedá-las. Quando o objetivo é adaptar um *web site* para que possa se beneficiar das

funcionalidades da aplicação, essa tarefa pode tornar-se cansativa, repetitiva e sujeita a erros, visto que seria necessária a atualização de todas as páginas HTML.

Para facilitar a conversão de um objeto de aprendizagem Web, foi desenvolvida uma aplicação que realiza algumas alterações automaticamente, adaptando as páginas ao formato requerido pela aplicação servidora e reduzindo o esforço de conversão. A aplicação, chamada *Site Adapter* (Figura 28), foi desenvolvida para que seja capaz de converter as páginas acessíveis a partir de um ponto inicial, mesmo que indiretamente (navegando através dos *links*).

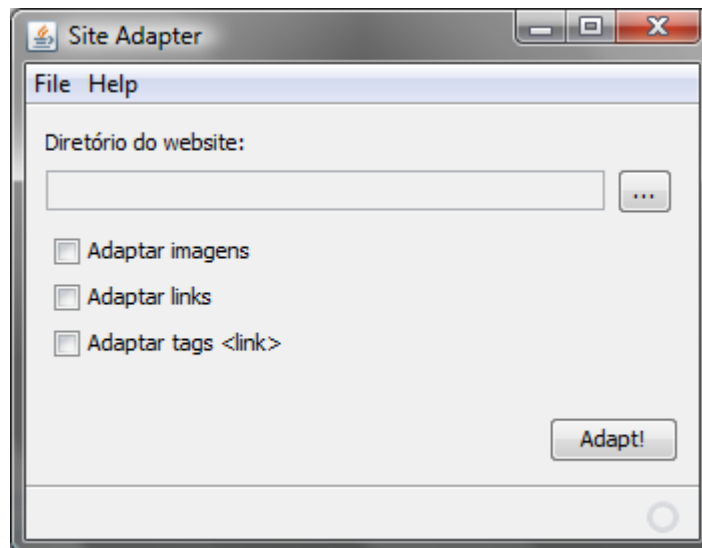


Figura 28. Tela inicial do *Site Adapter*.

Neste momento, o *Site Adapter* fez apenas algumas adaptações simples nos *web sites* desenvolvidos, como renomear arquivos, adicionar alguns elementos necessários, adaptar *links* e imagens (VARELLA, 2009-a). Porém, como trabalho futuro, pretende-se expandir sua utilização, possibilitando facilitar conversão dos sites existentes na Web para páginas interoperáveis entre Web, Móveis e TV digital.

### 4.3 Conjuntos de Metadados de Extensão aos Padrões Educacionais

Uma vez apresentadas as recomendações para criação de objetos de aprendizagem interoperáveis e o *framework Interop*, o objetivo desta seção é descrever a proposta de metadados de extensão aos padrões educacionais. Apresenta, assim, algumas categorias de metadados ainda não previstos pelos padrões educacionais e que, além de importantes devido à inserção das novas mídias de TV digital e móveis, possibilitam a interoperabilidade dos aplicativos entre as diferentes plataformas.

O IEEE LOM é a base dos padrões de metadados utilizados para a educação e, por esse motivo, também foi utilizado como referência neste trabalho, uma vez que o principal objetivo é a aplicação da proposta na área educacional.

As seções a seguir apresentam os dois conjuntos de metadados propostos com extensão ao padrão LOM, juntamente com o motivo da recomendação de cada um dos mesmos.

### 4.3.1 Metadados de Adaptação

Embora a situação ideal fosse aquela em que qualquer conteúdo ou mídia pudesse ser transmitido e executado exatamente da mesma forma em qualquer tipo de dispositivo, algumas características específicas de cada um deles impedem que esse ideal seja alcançado.

Tomando como exemplo um curso educacional em vídeo, o mesmo deverá ser disponibilizado em pelo menos dois formatos (dois arquivos distintos): um para dispositivos móveis (com codificação H.263) e outro para a Web e TV digital (com codificação H.264). Porém, embora sejam duas mídias separadas, o conteúdo é o mesmo, apenas em formatos diferentes. Dessa forma, ambas as mídias terão o mesmo título, descrição, assunto, autor, palavras chaves, etc., ou seja, são referentes ao mesmo objeto de aprendizagem.

Se essas mídias forem documentadas como objetos de aprendizagem separados, pelo menos dois problemas poderão ser observados: (a) o trabalho de geração dos metadados do OA será dobrado, pois mesmo com todos os dados descritivos sendo iguais, será necessário documentar repetidas vezes, uma para cada arquivo físico; (b) quando o usuário procurar pelo assunto do curso, os dois vídeos serão apresentados para o usuário e o ele precisará selecionar o vídeo adequado, dependendo do tipo de dispositivo que está utilizando.

Uma melhor solução é catalogar de forma unificada esse conteúdo e quando o usuário selecionar tal objeto, que o sistema identifique automaticamente o tipo de dispositivo e envie a mídia adequada, sem que o usuário necessite saber dessa diferenciação.

Para suprir essa necessidade, nesta dissertação está sendo proposta uma extensão de metadados para adaptação, permitindo que os metadados técnicos possam ser diferenciados para cada plataforma alvo, enquanto que os demais dados descritivos (título, descrição, etc.) serão documentados uma única vez.

Para adoção junto ao LOM, ao invés de modificar metadados e sintaxe já existentes, optou-se por estender o grupo *4:Technical*, mantendo assim a compatibilidade com o padrão, conforme recomendação do próprio LOM. São metadados já existentes nesse grupo (IEEE, 2002): *4.1:Format*, *4.2:Size*, *4.3:Location*, *4.4:Requirement*, *4.4.1:OrComposite*, *4.4.1.1:Type*, *4.4.1.2:Name*, *4.4.1.3:MinimumVersion*, *4.4.1.4:MaximumVersion*, *4.5:InstallationRemarks*, *4.6:OtherPlatformRequirements* e *4.7:Duration*. Os metadados propostos como extensão foram numerados de *4.8* em diante, não conflitando com os pré-existentes.

O Apêndice B apresenta a especificação completa dos metadados sugeridos como extensão aos já existentes grupo *4:Technical* do LOM, os quais são explicados a seguir:

- *4.8:SupportedPlatforms*: indica em quais plataformas o objeto de aprendizagem está apto a ser executado. Foram previstos três tipos básicos de plataformas digitais para disponibilização de OAs: Web, DTV (*digital television* - para TV digital) e *Mobile* (para dispositivos móveis);
- *4.9:PlatformSpecificFeatures*: Conjunto de metadados que identifica as informações técnicas das mídias, da mesma forma que o item *4:Technical*, porém, aplicados a cada uma das diferentes plataformas para as quais o objeto de aprendizagem foi previsto. Esse novo grupo de metadados é opcional, ou seja,

pode ser omitido caso os mesmos metadados técnicos gerais informados se apliquem a todas as plataformas previstas (metadados 4.1 a 4.7 do LOM), mas pode ser repetido mais de uma vez, caso o valor de um ou mais metadados técnicos sejam específicos (diferentes) para determinada plataforma indicada no item 4.8:*SupportedPlatforms*;

- 4.9.1:*SpecificPlatform*: Tipo da plataforma digital à qual se aplicam os parâmetros. Utiliza o mesmo vocabulário de tipos de plataforma usado no item 4.8:*SupportedPlatforms*;
- 4.9.2:*SpecificFormat*: Formato da mídia criada para utilização na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.1:*Format* do LOM, porém, aplicadas à mídia específica;
- 4.9.3:*SpecificSize*: Tamanho da mídia criada para utilização na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.2:*Size* do IEEE LOM, porém, aplicadas à mídia específica;
- 4.9.4:*SpecificLocation*: Uma sequência de caracteres utilizada para acessar a mídia criada especificamente para utilização na plataforma especificada no item 4.9.1:*SpecificPlatform*. Segue as mesmas definições e regras do item 4.3:*Location*, porém, aplicadas à mídia específica;
- 4.9.5:*SpecificRequirement*: Capacidades técnicas necessárias na plataforma definida no item 4.9.1 para utilizar essa mídia específica. Segue as mesmas definições e regras do item 4.4:*Requirement*, porém, aplicadas à mídia específica;
- 4.9.5.1:*SpecificOrComposite*: Agrupamento de múltiplos requisitos para a mídia na plataforma específica definida no item 4.9.1. Segue as mesmas definições e regras do item 4.4.1:*OrComposite*, porém, aplicadas à plataforma específica;
- 4.9.5.1.1:*SpecificType*: O tipo de tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.1:*Type*, porém, aplicadas à plataforma específica;
- 4.9.5.1.2:*SpecificName*: O nome da tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.2:*Name*, porém, aplicadas à plataforma específica;
- 4.9.5.1.3:*SpecificMinimumVersion*: Versão mínima da tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.3:*MinimumVersion*, porém, aplicadas à plataforma específica;
- 4.9.5.1.4:*SpecificMaximumVersion*: Maior versão aceitável da tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.4:*MaximumVersion*, porém, aplicadas à plataforma específica;
- 4.9.6:*SpecificInstallationRemarks*: Instruções de instalação do objeto de aprendizagem na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.5:*InstallationRemarks*, porém, aplicadas à plataforma específica;
- 4.9.7:*SpecificOtherPlatformRequirements*: Informações sobre outros requisitos de software e hardware necessários na plataforma definida no item 4.9.1. Segue

as mesmas definições e regras do item 4.6:*OtherPlatformRequirements*, porém, aplicadas à plataforma específica;

Como se pode observar, os nomes adotados para os metadados 4.9.2:*SpecificFormat* até 4.9.7:*SpecificOtherPlatformRequirements* seguiram a mesma nomenclatura dos itens correspondentes no LOM, porém, precedidos do prefixo *Specific*. O objetivo de manter nomenclatura similar é facilitar a identificação dos campos correspondentes aos do item 4:*Technical* geral (ou seja, independente da plataforma) com o grupo 4.9 *PlatformSpecificFeatures* (dados técnicos específicos para uma determinada plataforma). O prefixo *Specific* foi utilizado para não haver duplicidade de nomes, conforme recomendação do próprio padrão LOM.

Mesmo com a padronização de nomenclatura adotada neste trabalho para os metadados de adaptação, observa-se que a proposta é facilmente adaptável para estender os demais padrões de metadados já existentes e que ainda não suportam esse recurso.

### 4.3.2 Metadados para Segmentação

No uso de um objeto de aprendizagem, nem sempre um usuário deseja estudá-lo por completo. Em algumas situações o interesse é apenas por um ou alguns trechos relacionados a um determinado tema. Para que isso possa ser viabilizado, é necessário utilizar metadados de segmentação, que permitem subdividir logicamente um objeto de aprendizagem, permitindo a organização do mesmo por módulos ou por assuntos que o mesmo trata.

Um segmento é um fragmento contínuo de um item. Um segmento particular pode pertencer a um único objeto, mas pode ser membro de vários grupos de segmentos. Um grupo de segmentos denota uma coleção de segmentos que são associados por uma finalidade particular ou devido a uma propriedade em comum. Um grupo de segmentos pode conter segmentos ou outros grupos de segmentos (TVA, 2003-b).

Em função dessa necessidade, neste trabalho, o padrão LOM foi também estendido para fornecer suporte a segmentação. Para isso, um novo grupo de metadados foi criado, o 11:*SegmentInformationTable*, cujos metadados básicos originaram-se do padrão *TV-Anytime* (TVA, 2003-b), mas simplificados e adaptados para atender as necessidades educacionais. Com as modificações, esse conjunto de metadados possibilita segmentar, além de áudio e vídeo, outros formatos como texto, imagens ou até itens não digitais (como livros ou revistas).

O Apêndice C apresenta a especificação da lista de metadados de segmentação sugeridos, os quais são explicados a seguir:

- *11:SegmentInformationTable*: Novo grupo que conterá o conjunto de informações referente a segmentação e grupos de segmentos dos objetos de aprendizagem.
- *11.1:SegmentList*: Lista de segmentos de um objeto.
- *11.1.1:SegmentInformation*: Agrupamento das informações de um segmento.
- *11.1.1.2:SegmentIdentifier*: Identificador único do segmento no objeto de aprendizagem.
- *11.1.1.3:SegmentTitle*: Título do segmento.
- *11.1.1.4:SegmentDescription*: Descrição do conteúdo do segmento.

- *11.1.1.5:SegmentKeyword*: Palavras-chave referentes ao segmento.
- *11.1.1.6:SegmentMediaType*: Classifica o segmento em *document* (conteúdo texto, nos formatos txt, doc, odt ou não digital), *hyperdocument* (hiperdocumentos no formato HTML ou XHTML), *audio*, *video*, *image* ou *others* (outros).
- *11.1.1.7:Start*: indica o início do segmento no objeto de aprendizagem. Para tipo de mídia do segmento *document*, o início do segmento será a página e, opcionalmente, a linha de início (por exemplo, “Page10Line15”). Se o tipo de mídia for *hyperdocument*, será o nome de uma página, uma seção ou *bookmark* da página, indicado pelo caractere “#”, ou ainda alguma mídia embutida na página (nesse caso, o segmento é a seção ou a própria mídia referenciada). Se o tipo de mídia for *audio* ou *video*, será informado o tempo em que inicia o segmento. No caso de imagens, a linha e coluna de início do segmento.
- *11.1.1.8:End*: Localiza o final do segmento no objeto de aprendizagem. Para mídias do tipo *document*, *áudio*, *vídeo* ou *imagens*, segue a mesma sintaxe do item *11.1.1.8:Start*. Caso não seja especificado, o final do segmento será o final da mídia (digital ou impressa). Para os objetos do tipo *hyperdocument*, esse metadado não será preenchido, pois a seção indicada por *11.1.1.8:Start* já indica por si só seu início e fim.
- *11.2:SegmentGroupList*: Lista dos grupos de segmento.
- *11.2.1:SegmentGroupInformation*: Conjunto de informações de um grupo de segmentos.
  - *11.2.1.1:Identifier*: Identificador único do grupo de segmentos.
  - *11.2.1.2:GroupType*: Tipo de agrupamento. Terá o seguinte domínio de valores: *highlights* (destaques), *bookmarks* (marcações), *themeGroup* (tema), *preview* (resumo), *activities* (atividades) ou *other* (outros tipos de grupos);
  - *11.2.1.3:Title*: Título do grupo de segmentos.
  - *11.2.1.4:Description*: Descrição do conteúdo grupo de segmentos.
  - *11.2.1.5:Keyword*: Palavras-chave referentes ao grupo de segmentos.
  - *11.2.1.6:Segments*: Lista dos identificadores de segmentos que fazem parte do grupo.
    - *11.2.1.6.1:Identifier*: Identificador único do segmento (ou outro grupo de segmentos) que compõe o grupo.

Da mesma forma que os metadados de adaptação, pode-se observar que o grupo de metadados de segmentação também é flexível e pode ser facilmente transportado para outros padrões.

Portanto, foram apresentados os dois conjuntos de metadados propostos como extensão aos padrões educacionais existentes, aplicados neste caso ao IEEE LOM. Esses metadados ampliam as funcionalidades existentes nos padrões adicionais, permitindo referenciar objetos de aprendizagem projetados para diferentes plataformas, assim como os subdividir logicamente em partes ou temas de interesse.



## 5 IMPLEMENTAÇÕES PARA VALIDAR A PROPOSTA

A fim de verificar a viabilidade técnica das propostas de desenvolvimento interoperável deste trabalho, selecionou-se alguns objetos de aprendizagem e os implementou utilizando as recomendações efetuadas. Adicionalmente, cada um deles foi catalogado com as extensões de metadados técnicos e de segmentação propostas nesta dissertação, a fim de demonstrar sua utilização e sua aplicabilidade prática.

Dessa forma, inicialmente este capítulo apresenta a metodologia utilizada para escolha desses objetos de aprendizagem e as ferramentas e ambiente de validação adotado. A seguir, apresenta a implementação, telas de execução em cada plataforma e os metadados associados.

### 5.1 Metodologia para Escolha dos Objetos de Aprendizagem

Para escolha dos objetos de aprendizagem a serem convertidos, partiu-se da busca de quais eram os tipos de mídia mais recorrente em repositórios. Em seguida, com base nos dados levantados e com apoio dos grupos de pesquisa do CINTED (Centro de Novas Tecnologias na Educação), também participantes do projeto OBAA, selecionou-se alguns objetos de aprendizagem que contivessem os formatos de mídia identificados e que, ao mesmo tempo, atendessem aos requisitos pedagógicos.

Partindo disso, escolheu-se a Federação de Repositórios Educa Brasil<sup>14</sup> (FEB) como base para identificação dos principais tipos de mídias que aparecem nos objetos de aprendizagem. Essa base foi escolhida porque a mesma indexa mais de 50 mil objetos de aprendizagem provenientes de diversos repositórios como, por exemplo, o LUME<sup>15</sup> (Repositório Digital da Universidade Federal do Rio Grande do Sul), CESTA<sup>16</sup> (Coletânea de Entidades de Suporte ao Uso de Tecnologia na Aprendizagem), BIOE<sup>17</sup> (Banco Internacional de Objetos Educacionais) e BNDIGITAL<sup>18</sup> (Biblioteca Nacional Digital Brasil). Com isso, acredita-se que a pesquisa contemple a diversidade necessária para que a pesquisa retorne um resultado satisfatório, visto o objetivo da consulta.

A Tabela 5.1 exibe o resultado da busca, apresentando os principais formatos de mídia encontrados e sua quantidade. Cabe observar que nem todos os objetos de aprendizagem presentes no repositório possuíam o metadado contendo o formato

<sup>14</sup> Disponível em: <<http://feb.ufrgs.br/>>. Acesso em: out. 2010.

<sup>15</sup> Disponível em: <<http://www.lume.ufrgs.br/>>. Acesso em: out. 2010.

<sup>16</sup> Disponível em: <<http://www.cinted.ufrgs.br/CESTA/>>. Acesso em: out. 2010.

<sup>17</sup> Disponível em: <<http://objetoseducacionais2.mec.gov.br/>>. Acesso em: out. 2010.

<sup>18</sup> Disponível em: <<http://bndigital.bn.br/>>. Acesso em: out. 2010.

preenchido no momento da pesquisa. Mesmo assim, os que possuíam (total de 25.438 objetos de aprendizagem ou aproximadamente 42%) foram considerados como uma amostra suficiente, tendo em vista que a análise necessária para esta dissertação era apenas determinar os formatos de mídia presentes nos repositórios.

Tabela 5.1. Tipos de Mídia Existentes no FEB.

<b>Tipo de mídia</b>	<b>Quantidade de OAs</b>
PDF	23.285
Imagem	2.781
Flash	177
Hipertexto	142
Vídeo	126
Áudio	25
Outros	18

Percebe-se, pelos dados apresentados, que a grande maioria dos objetos presentes no FEB está no formato PDF (23.285 objetos de aprendizagem). Como já descrito anteriormente, esse formato é suportado pela Web e *smartphones* mais recentes, mas não está previsto para TV digital. A solução atual para tornar tais documentos interoperáveis também para essa plataforma seria converter o conteúdo texto para XHTML e as figuras para JPEG ou PNG. Como solução futura, poderia ser desenvolvido e integrado um leitor de PDF nessa plataforma.

O segundo tipo de mídia mais recorrente são imagens, em sua maioria já no formato JPEG (que é o recomendado nesta dissertação). As que ainda não estão nesse formato, ou em formato PNG, podem também ser facilmente convertidas para esses formatos com auxílio de um editor básico de imagens.

Animações em Flash aparecem em terceiro lugar. Na plataforma Web e de dispositivos móveis há suporte para esse tipo de conteúdo, embora nessa última de forma reduzida (*Flash Lite*). Já na TV digital, da mesma forma que para o formato PDF, o *Flash* também não possui suporte. Nesse caso, em pesquisa realizada pelo grupo pedagógico, observou-se que a maioria das animações nesse formato não possui interatividade com usuário, e nesse caso poderiam ser facilmente convertidas para um dos formatos de vídeo sugeridos nesta dissertação utilizando, por exemplo, um programa como o SUPER (ERIGHTSOFT, 2010). Já animações com interatividade, precisariam ser convertidas para utilizar também uma linguagem *script*, como o Lua, para serem acessíveis também a partir da TV digital.

Em quarto lugar, são listados os objetos de aprendizagem em hipertexto (HTML e XHTML), os quais uma vez padronizados para XHTML (e algumas adaptações por ventura necessárias) poderiam ser acessados de forma interoperável pelas três plataformas.

Na sequência, aparecem os objetos em formato de áudio e vídeos. Os que ainda não estão nos formatos recomendados nesta dissertação podem ser facilmente convertidos, utilizando ferramentas como o SUPER (ERIGHTSOFT, 2010).

Uma vez identificados os principais tipos de mídia, esses dados foram repassados aos grupos de pesquisa em educação participantes do projeto OBAA, que selecionaram

alguns objetos de aprendizagem que contivessem os formatos de mídia identificados e que, ao mesmo tempo, atendessem aos requisitos pedagógicos elencados pelos grupos de pesquisa da área da educação.

Como resultado desse trabalho, escolheu-se inicialmente dois objetos de aprendizagem do repositório CESTA. O primeiro é o “De onde vem a televisão?”, que conta a história da televisão e é composto por áudio e vídeo. Trata-se de um objeto de aprendizagem simples e adequado para primeira implementação. O segundo escolhido foi o “Outras Infâncias” que é um hiperdocumento que fala dos diversos aspectos da infância e adicionalmente ao anterior, além do áudio e vídeo, contém também texto e imagens.

Esses objetos de aprendizagem foram escolhidos por três motivos principais: (a) estarem adequados aos requisitos pedagógicos; (b) estarem de acordo com os principais formatos identificados, conforme análise realizada, e cuja solução pode assim ser replicada para inúmeros outros objetos de aprendizagem; (c) por estar em repositório de domínio da UFRGS (possibilitando acesso ao código fonte e mídias desses objetos de aprendizagem).

Foi também implementado um objeto de aprendizagem totalmente novo, o “Viva Saudável”, já projetado seguindo as recomendações de interoperabilidade. Adicionalmente aos anteriores, esse objeto de aprendizagem contém também alguns exercícios, importantes para validar a questão de interatividade com usuário utilizando as linguagens *script* recomendadas.

Mais detalhes sobre esses objetos de aprendizagem são fornecidos mais adiante, na seção correspondente a descrição da implementação de cada um deles. Antes disso, na próxima seção, será apresentado o ambiente utilizado para validação dos mesmos.

## 5.2 Ambiente de Validação e Ferramentas de Conversão de Formatos

Em relação ao ambiente de validação desses objetos de aprendizagem, no lado cliente, utilizou-se para a simulação na TV digital o Ginga-NCL *Virtual Set-top Box* (versão 0.10.1), que é a implementação de referência do Ginga-NCL disponível no Portal do Software Público Brasileiro (BRASIL, 2010).

Para validação nos dispositivos móveis, utilizou-se os celulares Motorola v196 e Nokia E.51 e os *smartphones* BlackBerry, Motorola Q11, Samsung GT-S8000B e iPhone, que eram os dispositivos disponíveis durante o desenvolvimento dos objetos de aprendizagem. Mas com essa gama de aparelhos em que os objetos de aprendizagem foram testados, acredita-se que os mesmos funcionem corretamente na grande maioria dos dispositivos móveis disponíveis no mercado.

Para a Web, usou-se o Mozilla Firefox, versão 3.0.5, por ser software livre, distribuído gratuitamente e conhecido por ter suas funcionalidades implementadas conforme os padrões W3C. Além disso, utilizou-se o Internet Explorer, que é o navegador mais utilizado atualmente, na sua versão atual (versão 8.0).

No lado servidor, foi utilizado o *framework Interop*, também implementado neste trabalho e descrito na Seção 4.2, no servidor de aplicação Glassfish 6.5 (SUN MICROSYSTEMS, 2010-h) para processamento das páginas web e o *Darwin Streaming Server* (APPLE, 2010-a) foi utilizado como servidor para *streaming* de vídeo. Para conversão de formatos de mídia, utilizou-se a ferramenta SUPER

(ERIGHTSOFT, 2010), que é uma ferramenta gratuita e a mais completa encontrada em termos de formatos suportados.

A descrição detalhada das ferramentas estudadas para compor o servidor de aplicação, além do motivo da escolha de cada uma delas, está detalhada no Apêndice D. A seguir, são apresentados os objetos de aprendizagem desenvolvidos, juntamente com os metadados propostos.

### 5.3 Conversão do Objeto de Aprendizagem “De onde vem a televisão?”

O primeiro objeto de aprendizagem convertido e catalogado com os metadados propostos é o vídeo “De onde vem a televisão”, disponível originalmente no repositório CESTA<sup>19</sup> e conta a origem da televisão e um pouco de sua história, além de explicar como é o funcionamento da transmissão do sinal de televisão atualmente. Apresenta também ao aluno um contexto amplo sobre o funcionamento dessa importante mídia.

No formato original, esse objeto de aprendizagem se encontra no formato *wmv*, com resolução 320x240. Portanto, esse vídeo necessitou ser convertido para os formatos recomendados para ser executado nas diferentes plataformas.

Assim, o vídeo foi convertido primeiramente para o formato H.264/AAC e encapsulado num arquivo *mp4*, contemplando a execução do mesmo na Web e TV digital. Para TV digital, foi necessária ainda a criação de um arquivo NCL (no momento, ainda de forma manual), o qual é necessário como base para execução do vídeo no ambiente de testes disponível (o *Ginga NCL Virtual Set-top Box*).

Na sequência, o vídeo foi convertido para a codificação H.263/AAC, em formato de arquivo *3gp*, para possibilitar execução nos dispositivos móveis. Logo, o vídeo é disponibilizado em dois formatos distintos, que permitem a execução do conteúdo nos três ambientes de testes.

Embora esse objeto possa ser acessado diretamente a partir de *players* de vídeo dos dispositivos (como o *Real Player* e *Quick Time*, por exemplo), para tornar ainda mais transparente o acesso a esse objeto de aprendizagem, os vídeos foram embutidos em páginas XHTML, utilizando o *framework Interop*. Dessa forma, o usuário utiliza sempre o mesmo endereço de acesso, tanto através de navegadores Web a partir de computador pessoal como a partir de dispositivos móveis. O trecho de código fonte referente ao vídeo é apresentado na Figura 29.

Como se pode observar, a *tag* `<i:video>` referencia dois endereços: o primeiro (apontado pela propriedade *mobileHref*) indica o endereço físico do vídeo para dispositivos móveis, enquanto que o segundo (apontado pela propriedade *webHref*) indica o endereço para acesso via *browsers* a partir de computadores pessoais. Com isso, quando a página é requisitada, o servidor de aplicação identifica automaticamente o tipo de dispositivo e retorna a página adequada para a plataforma.

---

<sup>19</sup> Disponível em: <http://cesta.cinted.ufrgs.br/sacca/player/dove/dove7.wmv>. Acesso em: nov. 2009.

```

16 <body>
17   <div id="conteudo">
18     <i:video width="640" height="496"
19       mobileHref="rtsp://143.54.132.100/dove7.3gp"
20       webHref="rtsp://143.54.132.100/dove7.mp4"
21       autoplay="true">
22       Assista ao Video
23     </i:video>
24   </div>
25 </body>

```

Figura 29. Trecho do código fonte do objeto de aprendizagem “De onde vem a televisão?”.

A Figura 30 apresenta um trecho da página XHTML gerada a partir do acesso por meio de um *browser* no computador pessoal, enquanto que a Figura 31 apresenta um trecho da página retornada quando acessada via dispositivo móvel. No primeiro caso, o vídeo é exibido no próprio *browser*, por meio de um *plugin* do *Quick Time*, enquanto que no segundo caso, é disponibilizado um *link* para que o vídeo seja exibido no *player* padrão do dispositivo móvel.

```

<body>
  <div id="conteudo">
    <script src="AC_Quicktime.js" language="JavaScript" type="text/javascript"></script>
    <script language="JavaScript" type="text/javascript">
      QT_WriteOBJECT('', '640', '512', '',
        'qtsrc', 'rtsp://143.54.132.100/dove7.mp4',
        'TYPE', 'video/quicktime',
        'autohref', 'false',
        'autoplay', 'true',
        'controller', 'true',
        'target', 'myself');
    </script>
  </div>
</body>

```

Figura 30. Código XHTML gerado pelo *Interop* a partir do acesso por PCs.

```

<body>
<div id="conteudo">
  <div>
    <a href="rtsp://143.54.132.100/dove7.3gp">Assista ao Video</A>
  </div>
</div>
</body>

```

Figura 31. Código XHTML gerado pelo *Interop* a partir do acesso por dispositivos móveis.

Já para TV digital, conforme já citado anteriormente, um arquivo NCL deverá existir para executar o vídeo, conforme trecho de código fonte ilustrado na Figura 32. A *tag* “*port*” indica que o vídeo representado pelo identificador *vídeo* iniciará assim que a aplicação for executada. Além disso, são criados dois *links* que permitem parar e reiniciar o vídeo com os botões interativos de cor vermelha e azul do controle remoto, respectivamente.

```

<body>
  <!-- Nodo inicial do documento -->
  <port id="pInicio" component="video"/>

  <!-- mídia do vídeo -->
  <media id="video" type="video/mpeg" src="media/dove_mp4.mp4" descriptor="dTelaCheia"/>

  <!-- ao clicar no botão vermelho, pausa o vídeo -->
  <link id="onSelCursorRedVideo" xconnector="connBase#onKeySelectionStartNStopN">
    <bind component="video" role="onSelection">
      <bindParam name="keyCode" value="RED" />
    </bind>
    <bind component="video" role="stop"/>
  </link>

  <!-- ao clicar no botão azul, reinicia o vídeo -->
  <link id="onSelCursorBlueVideo" xconnector="connBase#onKeySelectionStartNStopN">
    <bind component="video" role="onSelection">
      <bindParam name="keyCode" value="BLUE" />
    </bind>
    <bind component="video" role="start"/>
  </link>
</body>

```

Figura 32. Código NCL para execução do OA na TV digital.

Com isso, o objeto de aprendizagem pode ser apresentado nas plataformas Web, móvel e de TV digital. A Figura 33 ilustra a apresentação dos objetos de aprendizagem no *browser* Mozilla Firefox, no *smartphone* BlackBerry e no *Ginga NCL Virtual Set-top Box*, respectivamente.



Figura 33. Objeto de Aprendizagem “De Onde vem a TV?”, apresentado nas três plataformas.

Em relação aos metadados, a Figura 34 ilustra os valores gerados para o grupo 4: *Technical*, já com a extensão proposta. Como se pode observar, o novo metadado *SupportedPlatforms* indica que este objeto de aprendizagem está disponível para execução nas plataformas móvel, Web e de TV digital.

O formato padrão do objeto (metadado *Format*) é *vídeo/mpeg*, cujo acesso pode ser realizado pela URL *rtsp://143.54.132.100/dove7.mp4* (metadado *Location*). Para executar o objeto, nesse formato padrão, se faz necessário o Mozilla Firefox a partir da versão 3.0, além de um *plugin* de vídeo (metadados do grupo *Requirement* e *OtherPlatformRequirement*). Cabe ressaltar, que esses metadados (*Format*, *Location*, *Requirement* e *OtherPlatformRequirement*) já fazem parte do padrão IEEE LOM,

enquanto que o metadado *SupportedPlatform*, assim como os citados a seguir, fazem parte da extensão proposta nesta dissertação.

O grupo *PlatformSpecificFeatures* aparece duas vezes, uma para móveis e uma para TV digital (indicado pelo metadado *PlatformType*). Para a plataforma móvel, descreve que o vídeo possui um formato, localização e requisitos específicos (indicado pelos metadados *SpecificFormat*, *SpecificLocation* e *SpecificOtherPlatformRequirements*, respectivamente).

Já para TV digital, apenas especializa os metadados referentes aos requisitos específicos (*SpecificRequirements* e subitens), indicando que na TV digital necessita do *middleware* Ginga para executar esse vídeo, enquanto que na Web será executado diretamente de um *browser*. Uma vez que as demais informações, como formato e localização, são as mesmas dos metadados gerais (*Format* e *Location*, respectivamente), pode-se omiti-los, pois herda os valores informados no metadado geral equivalente (o de mesmo nome, porém, sem o prefixo *Specific*).

```
<Technical>
  <SupportedPlatforms>Mobile</SupportedPlatforms>
  <SupportedPlatforms>DTV</SupportedPlatforms>
  <SupportedPlatforms>WEB</SupportedPlatforms>

  <Format>video/mpeg</Format>
  <Location>http://143.54.132.100:12000/dove.mp4</Location>
  <Requirement>
    <OrComposite>
      <Type>Browser</Type>
      <Name>Mozilla Firefox</Name>
      <MinimumVersion>3.0</MinimumVersion>
    </OrComposite>
  </Requirement>
  <OtherPlatformRequirements>Plugin para videos</OtherPlatformRequirements>

  <PlatformSpecificFeatures>
    <PlatformType>Mobile</PlatformType>
    <SpecificFormat>video/3gp</SpecificFormat>
    <SpecificLocation>http://143.54.132.100:12000/dove.mp3</SpecificLocation>
    <SpecificOtherPlatformRequirements>RealPlayer</SpecificOtherPlatformRequirements>
  </PlatformSpecificFeatures>

  <PlatformSpecificFeatures>
    <PlatformType>DTV</PlatformType>
    <SpecificRequirement>
      <SpecificOrComposite>
        <SpecificType>middleware</SpecificType>
        <SpecificName>ginga</SpecificName>
        <SpecificMinimumVersion>1.0</SpecificMinimumVersion>
      </SpecificOrComposite>
    </SpecificRequirement>
  </PlatformSpecificFeatures>
</Technical>
```

Figura 34. XML dos metadados de extensão ao grupo técnico.

Pode-se observar que o *framework Interop* (ou algum outro) poderia facilmente obter automaticamente a localização do vídeo a partir arquivo de metadados XML apresentado. Porém, na versão atual do *Interop*, essa integração ainda não foi finalizada e por esse motivo o endereço do arquivo para cada plataforma foi programada

diretamente no código fonte (conforme demonstrado na Figura 29). Mas essa integração poderá ser realizada em trabalhos futuros.

Já a Figura 35 ilustra o uso dos metadados de segmentação, catalogando dois segmentos do objeto de aprendizagem. O primeiro segmento (*SegmentIdentifier* “0001”), que inicia no segundo 46 (metadado *Start*) e finaliza no tempo 2 minutos e 32 segundos (*End*), conta a história da televisão (metadados *SegmentTitle* e *SegmentKeyword*), enquanto que o segundo segmento, que inicia no tempo 3 minutos e 29 segundos e termina nos 3 minutos e 53 segundos, fala sobre as novidades da TV digital.

```
<SegmentInformationTable>
  <SegmentList>
    <SegmentInformation>
      <SegmentIdentifier>0001</SegmentIdentifier>
      <SegmentTitle>História da Televisão</SegmentTitle>
      <SegmentKeyword>Televisão</SegmentKeyword>
      <SegmentKeyword>História</SegmentKeyword>
      <SegmentMediaType>video</SegmentMediaType>
      <Start>PT00H00M46S</Start>
      <End>PT00H02M32S</End>
    </SegmentInformation>
    <SegmentInformation>
      <SegmentIdentifier>0002</SegmentIdentifier>
      <SegmentTitle>Televisão Digital</SegmentTitle>
      <SegmentKeyword>Televisão</SegmentKeyword>
      <SegmentKeyword>Digital</SegmentKeyword>
      <SegmentMediaType>video</SegmentMediaType>
      <Start>PT00H03M29S</Start>
      <End>PT00H03M53S</End>
    </SegmentInformation>
  </SegmentList>
</SegmentInformationTable>
```

Figura 35. XML com dois segmentos do vídeo “De onde vem a televisão?”.

Esse objeto de aprendizagem convertido encontra-se disponível no Portal OBAA (OBAA, 2010) ou pode ser acessado diretamente pelo endereço <<http://gia.inf.ufrgs.br/OBAI/faces/storage/ondevemtv/index.jsp>>.

## 5.4 Conversão do Objeto “Outras Infâncias”

Realizada a validação do primeiro objeto de aprendizagem composto por um vídeo, foi escolhido um segundo objeto de aprendizagem chamado “Outras Infâncias”, desenvolvido pelo NUTED<sup>20</sup> (Núcleo de Tecnologia Digital Aplicada à Educação) da UFRGS. Esse foi escolhido por utilizar os textos e imagens, e foi escolhido por ser relativamente simples, sendo um bom estudo de caso para a validação inicial desses formatos de mídia, além de já ser validado pedagogicamente. A Figura 36 ilustra o site original, atualmente publicado na Web, quando acessado no módulo “Desafios”, página “Infâncias”.

<sup>20</sup> Disponível em: <<http://homer.nuted.edu.ufrgs.br/ei2007/infancias/index.html>>. Acesso em: ago. 2010.



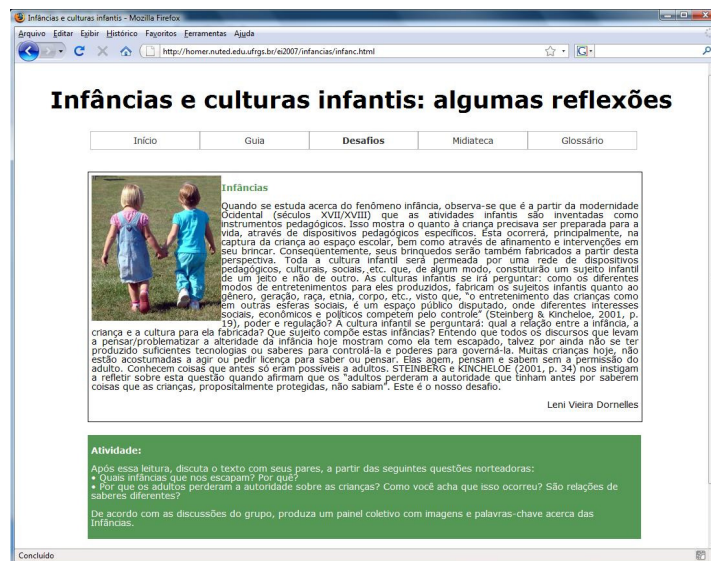


Figura 36. Objeto de aprendizagem “Outras Infâncias” no formato original (NUTED, 2009).

Conforme ilustrado, cada página do objeto de aprendizagem “Outras Infâncias” contém grande volume de texto em cada página, ultrapassando o que pode ser exibido em cada tela na TV. Na TV digital, embora seja prevista a propriedade de rolagem em NCL, esse recurso ainda não está disponível nos simuladores atuais. Além disso, não é muito confortável em uma plataforma em que os usuários estão acostumados a ter todo conteúdo visível na tela, precisarem rolar o texto para ler todo o texto.

Por isso, cada página do Outras Infâncias necessitou ser dividida. Tendo em vista manter o modelo pedagógico do objeto de aprendizagem, o trabalho de separação em páginas menores (reduzindo o conteúdo de cada página) e de associação de novas imagens às páginas criadas foi realizado pelo NUTED - Núcleo de Tecnologia Digital Aplicada à Educação (NUTED, 2010), grupo de pesquisa também integrante do projeto OBAA.

Da mesma forma, o menu da versão original, que utiliza *Java script*, também necessitou ser reformulado para garantir, além da funcionalidade nos três ambientes, o mesmo padrão de usabilidade e interface. Isso porque o *script* utilizado originalmente ainda não é suportado nos simuladores atuais para TV digital e nos dispositivos móveis. Essa reformulação utilizou-se de um botão, onde o usuário clica para voltar ao menu, de forma a melhor aproveitar o espaço de tela, além de botões de navegação para acesso entre as páginas de um mesmo grupo de conteúdo.

Uma vez reorganizado o conteúdo, foram geradas as novas páginas seguindo as recomendações para interoperabilidade citadas anteriormente, utilizando o formato XHTML para texto, CSS para adaptar a exibição e JPEG para as imagens. A criação da página XHTML foi realizada utilizando o *framework Interop*, cujos trechos principais de código fonte são apresentados na Figura 37.

```

<head>
...
<i:link rel="stylesheet" href="infancias.css" type="text/css"/>
...
</head>
<body>
...
<i:img id="img1" src="img/infancias.jpg" styleClass="imagem" />
<h2>Infâncias</h2>
<ul>
<li>A partir da modernidade Ocidental (séculos XVII/XVIII) que as atividades infantis são inventadas como instrumentos pedagógicos;</li>
<li>Isso mostra o quanto a criação precisava ser preparada para a vida, através de dispositivos pedagógicos específicos;</li>
<li>Toda a cultura infantil será permeada por uma rede de dispositivos pedagógicos, culturais, sociais, etc que, de algum modo, constituirão um sujeito infantil de um jeito e não de outro;</li>
</ul>
...
</body>

```

Figura 37. Trecho de código fonte do objeto de aprendizagem “Outras Infâncias”.

Como se pode observar, o corpo é composto por uma imagem (referenciada pela tag `<i:img ... />`) e texto, o qual será apresentado conforme a folha de estilo CSS retornada pelo *framework Interop* (referenciada pela tag `<i:link href="infancias.css" ... />`). As tags `<head>`, `<body>`, `<h2>`, `<ul>` e `<li>` fazem parte do próprio XHTML 1.1. Por questão de espaço e clareza na apresentação do código fonte, suprimiu-se a parte correspondente a apresentação do menu esquerdo e painel superior, mas é equivalente ao código apresentado.

A Figura 38 apresenta fotografias da execução desse objeto de aprendizagem na Web, em TV digital e no celular, respectivamente.

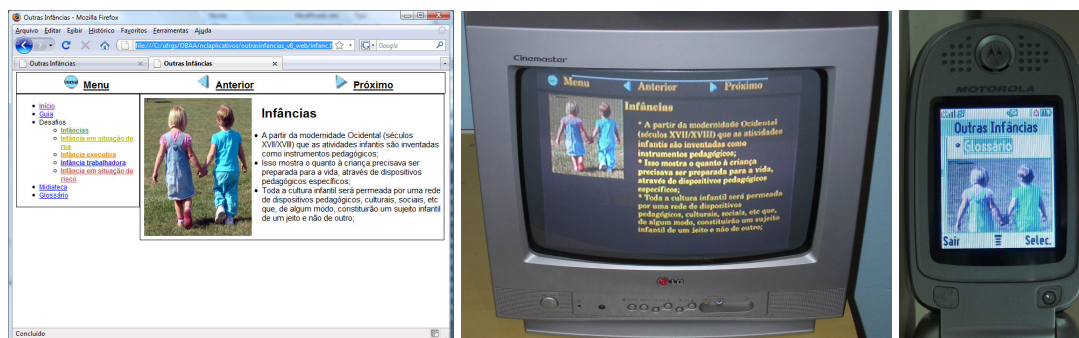


Figura 38. Objeto de Aprendizagem “Outras Infâncias” nas plataformas Web (esquerda), TV digital (meio) e móvel (direita).

Os arquivos XHTML permanecem iguais nas três plataformas. Porém, na TV digital, no *Ginga-NCL Virtual Set-top Box*, também foi necessária a criação de um documento NCL base, como o da Figura 39, pois ele indicará ao *player NCL* quais páginas serão exibidas. Nesta versão, o arquivo NCL foi gerado manualmente, mas em versões futuras do *framework Interop*, essa criação poderá ser automatizada tendo em vista facilitar ainda mais o desenvolvimento.

```

<media id="infanc1_titulo" type="text/html" src="infanc.html#titulo" descriptor="dTitulo" />
<media id="infanc1_menu" type="text/html" src="infanc.html#menu" descriptor="dConteudo"/>
<media id="infanc1_conteudo" type="text/html" src="infanc.html#conteudo" descriptor="dConteudo"/>

<media id="infanc2_titulo" type="text/html" src="infanc2.html#titulo" descriptor="dTitulo" />
<media id="infanc2_menu" type="text/html" src="infanc2.html#menu" descriptor="dConteudo"/>
<media id="infanc2_conteudo" type="text/html" src="infanc2.html#conteudo" descriptor="dConteudo"/>

<media id="infanc3_titulo" type="text/html" src="infanc3.html#titulo" descriptor="dTitulo" />
<media id="infanc3_menu" type="text/html" src="infanc3.html#menu" descriptor="dConteudo"/>
<media id="infanc3_conteudo" type="text/html" src="infanc3.html#conteudo" descriptor="dConteudo"/>

```

Figura 39. Trecho NCL onde os arquivos HTML são referenciados.

Em relação aos metadados, conforme ilustrado na Figura 40, o “Outras Infâncias” assim como o objeto anterior é suportado pelas três plataformas. Porém, para esse objeto de aprendizagem, a localização é a mesma para as plataformas Web e móvel, sendo que a adaptação é realizada pelo servidor de aplicação e *framework Interop*. Já para a TV digital foi disponibilizado um arquivo separado, agrupado no formato *zip*, para ser descompactado e transmitido via *broadcast* pelas emissoras ou até executado diretamente a partir de um *pen drive* em qualquer *set-top box* que tenha o *middleware Ginga*.

```

<Technical>
  <SupportedPlatforms>Mobile</SupportedPlatforms>
  <SupportedPlatforms>DTV</SupportedPlatforms>
  <SupportedPlatforms>WEB</SupportedPlatforms>

  <Format>XHTML</Format>
  <Format>JPEG</Format>
  <Location>http://gia.inf.ufrgs.br/OBAI/faces/storage/OutrasInfancias/index.jsp
  </Location>
  <Requirement>
    <OrComposite>
      <Type>Browser</Type>
      <Name>MS-Internet Explorer</Name>
    </OrComposite>
    <OrComposite>
      <Type>Browser</Type>
      <Name>Mozilla Firefox</Name>
    </OrComposite>
  </Requirement>

  <PlatformSpecificFeatures>
    <PlatformType>DTV</PlatformType>
    <Format>ZIP</Format>
    <Location>http://gia.inf.ufrgs.br/OBAI/storage/OutrasInfancias/outrasinfancias_tvd.zip
    </Location>
    <SpecificRequirement>
      <SpecificOrComposite>
        <SpecificType>middleware</SpecificType>
        <SpecificName>ginga</SpecificName>
        <SpecificMinimumVersion>1.0</SpecificMinimumVersion>
      </SpecificOrComposite>
    </SpecificRequirement>
  </PlatformSpecificFeatures>
</Technical>

```

Figura 40. XML com metadados técnicos do OA.

Em relação aos metadados de segmentação, aqui é exemplificado o uso dos metadados de agrupamento (neste caso por assunto), onde cada segmento é representado por uma página XHTML referente aquele assunto. A Figura 41 ilustra dois

segmentos (páginas `i_trab.jsp` e `i_trab2.jsp`) e o agrupamento dessas páginas (metadados *SegmentGroupInformation*, *Segments* e *Identifier*). Neste exemplo, foram descritos e agrupados apenas dois segmentos, mas podem ser incluídos tantos quantos forem desejáveis.

```
<SegmentInformationTable>
  <SegmentList>
    <SegmentInformation>
      <SegmentIdentifier>i_trab0001</SegmentIdentifier>
      <SegmentTitle>Infância Trabalhadora - Página 1</SegmentTitle>
      <SegmentMediaType>XHTML</SegmentMediaType>
      <Start>http://gia.inf.ufrgs.br/OBAI/faces/storage/OutrasInfancias/i_trab.jsp</Start>
    </SegmentInformation>
    <SegmentInformation>
      <SegmentIdentifier>i_trab0002</SegmentIdentifier>
      <SegmentTitle>Infância Trabalhadora-Página 2</SegmentTitle>
      <SegmentMediaType>XHTML</SegmentMediaType>
      <Start>http://gia.inf.ufrgs.br/OBAI/faces/storage/OutrasInfancias/i_trab2.jsp</Start>
    </SegmentInformation>
  </SegmentList>

  <SegmentGroupList>
    <SegmentGroupInformation>
      <Identifier>grp0001</Identifier>
      <GroupType>themeGroup</GroupType>
      <Title>Infância Trabalhadora</Title>
      <Segments>
        <Identifier>i_trab0001</Identifier>
        <Identifier>i_trab0002</Identifier>
      </Segments>
    </SegmentGroupInformation>
  </SegmentGroupList>
</SegmentInformationTable>
```

Figura 41. Exemplo dos metadados de segmentação.

Esse objeto de aprendizagem convertido também está disponível no Portal OBAA (OBAA, 2010) ou pode ser acessado diretamente no endereço `<http://gia.inf.ufrgs.br/OBAI/faces/storage/OutrasInfancias/index.jsp>`.

## 5.5 Implementação do Curso Educacional “Viva Saudável”

Com o sucesso nas conversões anteriores, partiu-se para uma implementação do curso educacional “Viva Saudável”, cuja interface já foi originalmente projetada de forma interoperável (BARBOSA, ROESLER e REATEGUI, 2009) e, adicionalmente aos anteriores, também com exercícios interativos desenvolvidos utilizando linguagens *script*.

Em relação à interoperabilidade, utilizaram-se os padrões recomendados. Para vídeo, o H.264/AVC, convertido para H.263 encapsulado em 3GP nos celulares. O áudio foi AAC-LC. O texto utilizou-se do XHTML e CSS. As imagens foram convertidas para JPEG com diferentes resoluções, adaptando o tamanho para os diferentes dispositivos.

A Figura 42 ilustra a página inicial do curso, que é o índice, sendo exibida na Web (a), no celular Nokia E51 (b) e na televisão por meio do *Virtual Set-top Box* (c), enquanto a Figura 43 ilustra uma página de conteúdo do curso, exibido nos mesmos dispositivos.

Pode ser observado que há algumas leves diferenças no leiaute, principalmente em relação à posição das imagens e tamanho das margens do corpo do índice, além de diferenças no tamanho e no tipo das fontes.



(a) Computador pessoal

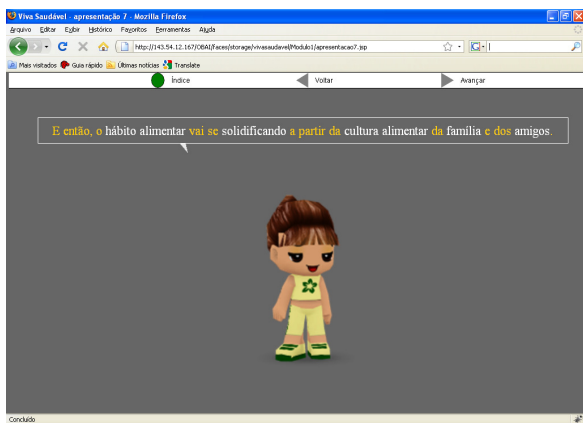


(b) Celular

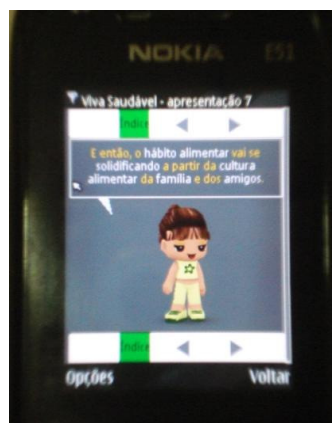


(c) TV Digital

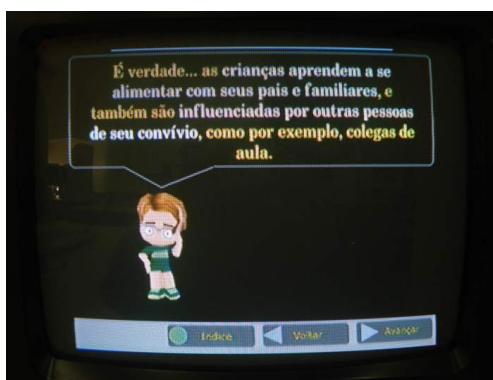
Figura 42. Página inicial do curso na Web, celular e TV digital.



(a) Computador pessoal



(b) Celular



(c) TV Digital

Figura 43. Exemplo de página de conteúdo do objeto de aprendizagem “Viva Saudável”.

Adicionalmente aos objetos de aprendizagem anteriores, o “Viva Saudável” possui também exercícios interativos, como o *Quis* ilustrado na Figura 44.

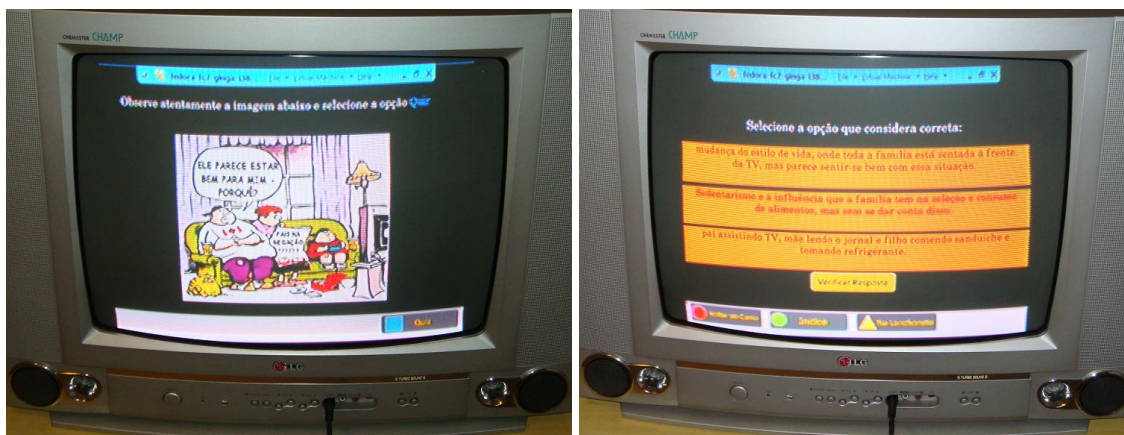


Figura 44. Exercícios interativos no Viva Saudável.

Em relação aos metadados, conforme ilustrado na Figura 45, o “Viva Saudável” é suportado pelas três plataformas e é documentado de forma bem semelhante ao “Outras Infâncias”, mudando apenas o endereço de acesso. A localização é a mesma para as plataformas Web e móvel, enquanto que para a TV digital foi disponibilizado um arquivo diferente, agrupado no formato *zip*, para ser descompactado e enviado via

*broadcast* ou executado diretamente a partir de um *pen drive* em qualquer *set-top box* que tenha o *middleware* Ginga.

Em relação aos metadados de segmentação, aqui é exemplificado o uso dos metadados de segmentação utilizados para reunir as atividades referentes ao módulo 1, onde cada segmento é representado por uma página XHTML referente aquele assunto. A Figura 46 ilustra três segmentos (*modulo01parafazer01*, *modulo01parafazer02* e *modulo01parafazer03*) e o agrupamento dessas páginas (metadados *SegmentGroupInformation*, *Segments* e *Identifier*).

```
<Technical>
<SupportedPlatforms>Mobile</SupportedPlatforms>
<SupportedPlatforms>DTV</SupportedPlatforms>
<SupportedPlatforms>WEB</SupportedPlatforms>

<Format>XHTML</Format>
<Format>JPEG</Format>
<Location>http://gia.inf.ufrgs.br/OBAI/faces/storage/VivaSaudavel/index.jsp</Location>
<Requirement>
  <OrComposite>
    <Type>Browser</Type>
    <Name>MS-Internet Explorer</Name>
  </OrComposite>
  <OrComposite>
    <Type>Browser</Type>
    <Name>Mozilla Firefox</Name>
  </OrComposite>
</Requirement>

<PlatformSpecificFeatures>
<PlatformType>DTV</PlatformType>
<Format>ZIP</Format>
<Location>http://gia.inf.ufrgs.br/OBAI/storage/VivaSaudavel/vivasaudavel_tvd.zip</Location>
<SpecificRequirement>
  <SpecificOrComposite>
    <SpecificType>middleware</SpecificType>
    <SpecificName>ginga</SpecificName>
    <SpecificMinimumVersion>1.0</SpecificMinimumVersion>
  </SpecificOrComposite>
</SpecificRequirement>
</PlatformSpecificFeatures>
</Technical>
```

Figura 45. XML com metadados técnicos do OA “Viva Saudável”.

```
<SegmentInformationTable>
  <SegmentList>
    <SegmentInformation>
      <SegmentIdentifier>modulo01parafazer01</SegmentIdentifier>
      <SegmentTitle>Para Fazer - Questionário</SegmentTitle>
      <SegmentMediaType>XHTML</SegmentMediaType>
      <Start>http://gia.inf.ufrgs.br/OBAI/faces/storage/VivaSaudavel/parafazer1.jsp</Start>
    </SegmentInformation>
    <SegmentInformation>
      <SegmentIdentifier> modulo01parafazer02</SegmentIdentifier>
      <SegmentTitle>Para Fazer - Calcular o IMC</SegmentTitle>
      <SegmentMediaType>XHTML</SegmentMediaType>
      <Start>http://gia.inf.ufrgs.br/OBAI/faces/storage/VivaSaudavel/calcoimo.jsp</Start>
    </SegmentInformation>
    <SegmentInformation>
      <SegmentIdentifier> modulo01parafazer03</SegmentIdentifier>
      <SegmentTitle>Para Fazer - Quiz</SegmentTitle>
      <SegmentMediaType>XHTML</SegmentMediaType>
      <Start>http://gia.inf.ufrgs.br/OBAI/faces/storage/VivaSaudavel/video1.jsp</Start>
    </SegmentInformation>
  </SegmentList>

  <SegmentGroupList>
    <SegmentGroupInformation>
      <Identifier>grp0001</Identifier>
      <GroupType>activities</GroupType>
      <Title>Atividades do Módulo 1</Title>
      <Segments>
        <Identifier>modulo01parafazer01</Identifier>
        <Identifier>modulo01parafazer02</Identifier>
        <Identifier>modulo01parafazer03</Identifier>
      </Segments>
    </SegmentGroupInformation>
  </SegmentGroupList>
</SegmentInformationTable>
```

Figura 46. Metadados de segmentação do OA “Viva Saudável”.

Portanto, observa-se que os metadados de adaptação permitem descrever as plataformas para as quais os objetos de aprendizagem foram previstos, assim como indicar endereço de acesso único ou diferenciado para cada plataforma. Permite ainda expressar os requisitos mínimos necessários (ou características específicas dentro de cada plataforma).

Foram ainda listadas utilizações dos metadados de segmentação, utilizando objetos de aprendizagem compostos por vídeos e páginas HTML, observando-se que os mesmos permitem uma documentação mais detalhada, permitindo o agrupamento por assuntos, módulos ou atividades (conforme exemplos), entre outros.

## 5.6 Conclusões sobre os Objetos de Aprendizagem Desenvolvidos

A partir dos objetos de aprendizagem desenvolvidos, observa-se que a criação de objetos de aprendizagem interoperáveis é viável, desde que sejam utilizados formatos de mídia de uso em comum, conforme as recomendadas nesta dissertação. Além disso, a utilização de um *framework* como o *Interop* facilita o desenvolvimento desses objetos de aprendizagem e, adicionalmente, fornece mecanismos que possibilitam adaptações visando obter o melhor uso dos recursos de cada plataforma.

Comparando o desenvolvimento interoperável ao tradicional, com base nas implementações realizadas, chegou-se as seguintes conclusões:

- **Facilidade de implementação:** quando comparado ao desenvolvimento tradicional, uma vez configurado o ambiente básico, observou-se que não há aumento de esforço na construção de objetos interoperáveis. Pelo contrário, reduz o tempo de desenvolvimento, uma vez que a criação interoperável permite que o objeto de aprendizagem implementado seja executado nas diferentes plataformas, ao invés de precisar fazer três implementações. Já o tempo necessário para os testes de execução nas três plataformas não diminui, pois mesmo com o desenvolvimento interoperável os testes se fazem necessários em cada dispositivo para o qual o mesmo foi previsto;
- **Qualidade da interface:** é possível criar interfaces atraentes e com boa usabilidade, conforme já demonstrado pelas implementações realizadas. Porém, caso seja desejado manter exatamente a mesma interface para as três plataformas, é necessário que a mesma seja projetada em função das plataformas com menos recursos, que é o caso da TV digital e dispositivos móveis. Pode assim haver alguma necessidade de limitação da quantidade conteúdo no ambiente Web para atender as demais plataformas. Pode ser observado, por exemplo, que no objeto de aprendizagem “Outras Infâncias” foi necessário reduzir a quantidade de texto por página e simplificar o menu utilizado em função da TV digital e dos dispositivos móveis. Mas, conforme apresentado nesse objeto, assim como no “Viva Saudável”, é possível criar interfaces interoperáveis tão atrativas quanto as de objetos de aprendizagem criados de forma tradicional;
- **Curva de aprendizagem:** mesmo utilizando as recomendações propostas, ainda assim o desenvolvedor precisa de conhecimentos específicos sobre cada um dos tipos de plataforma. Além disso, deve-se contar o tempo necessário para o estudo e assimilação das recomendações de interoperabilidade e o uso do



*framework Interop*. Devido aos fatos citados, o desenvolvimento tradicional apresenta uma curva de aprendizado menor.

- **Manutenção:** comparando o esforço necessário para efetuar alterações no conteúdo, bem como inserir e/ou remover trechos de conteúdo ou páginas inteiras. Utilizando a abordagem tradicional, a necessidade de uma alteração leva a três mudanças diferentes, uma para cada tipo de dispositivo alvo, ou seja, o tempo e o esforço são triplicados. Além disso, caso seja feita uma manutenção em apenas uma ou duas plataformas, o conteúdo fica inconsistente entre si, prejudicando quem utiliza a plataforma não atualizada.

Os detalhes sobre a obtenção desses resultados são apresentados em (VARELLA, 2009-a).

Já em relação aos metadados, observa-se que há um pequeno aumento na quantidade de informações a ser preenchidas quando são catalogados objetos de aprendizagem interoperáveis. Porém, caso não fosse adotada a solução proposta, todo conjunto de metadados LOM precisaria ser duplicado para cada versão do objeto (uma vez para cada plataforma), mesmo com a maioria dos metadados sendo não técnicos, contendo a mesma informação descritiva. Além disso, apenas as informações técnicas que diferirem entre as plataformas deverão ser especializadas, reduzindo ainda mais o esforço de preenchimento das informações.

Em relação aos metadados de segmentação, também há alguns metadados adicionais a serem informados. Mas uma vez informados, facilitará as buscas posteriores pelos trechos de interesse e aumentará significativamente a reutilização, que é um dos principais objetivos dos objetos de aprendizagem.

Além disso, como as extensões propostas não impactam nos metadados já existentes, não haverá trabalho adicional para os professores que não desejam prever mais do que uma plataforma para execução de seu objeto de aprendizagem ou que, da mesma forma, não desejam segmentá-lo.

Há ainda algumas limitações que foram observadas durante o desenvolvimento dos objetos de aprendizagem interoperáveis, a citar:

- Os formatos PDF e Flash ainda não são previstos pela TV digital e, embora presente em grande quantidade nos repositórios de OAs, hoje precisariam ser convertidos para um dos formatos recomendados. Como alternativa, poderão ser definidos *plugins* para dar suporte a esses formatos;
- Algumas mídias, como os vídeos, ainda precisam estar em dois formatos, um para celulares e outro para TV digital e Web. Com a atual evolução dos *smartphones* e adesão por parte dos fabricantes ao H.264, essa limitação deverá ser ultrapassada em breve;
- A TV digital precisa de um arquivo NCL como base para apresentação dos aplicativos, o qual ainda está sendo gerado manualmente. Como continuidade, o *framework Interop* poderá ser melhorado para gerar esse arquivo automaticamente a partir dos metadados e *links* dos próprios hiperdocumentos.

Mesmo com essas limitações, as recomendações e exemplos de objetos de aprendizagem interoperáveis desenvolvidos nesta dissertação podem ser utilizados como diretrizes e replicados para inúmeros outros objetos de aprendizagem, tornando-os acessíveis nas plataformas Web, móvel e de TV digital.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um estudo visando padronização de mídias e linguagens para obter interoperabilidade entre TV digital, Web e dispositivos móveis, além de uma proposta de metadados de extensão aos padrões educacionais.

No capítulo 2, apresentou-se o estudo dos padrões para cada uma das plataformas, elencando os recursos disponíveis, restrições e principais recomendações de usabilidade. Identificou-se que as plataformas com maiores limitações são as de TV digital e móvel. Julgou-se que um computador na Web pode se adaptar a qualquer padrão sem maiores problemas, bastando o comprometimento dos produtores de conteúdo com uma proposta interoperável.

Com base nesse estudo, concluiu-se que é possível gerar objetos de aprendizagem compatíveis com as três plataformas, utilizando como formatos em comum o XHTML para texto (juntamente com CSS e linguagem *script*), JPEG ou PNG para imagens e AAC para áudio. Para vídeo, optou-se por utilizar o MP4/H.264 para TV digital e Web e 3GP/H.263 para dispositivos móveis, por ainda ser o mais difundido nessa plataforma.

Essas recomendações foram descritas no capítulo 4 (na seção 4.1), o qual apresenta também o *Interop* (na seção 4.2), um *framework* desenvolvido para facilitar a criação de objetos de aprendizagem interoperáveis baseados nas recomendações acima e, adicionalmente, permitir a realização de adaptações entre as diferentes plataformas, quando necessário.

Também foi realizado um estudo comparativo acerca dos diferentes padrões de metadados educacionais, multimídia, Web e para TV digital, o qual foi detalhado no capítulo 3. Como base nesse estudo, verificou-se a deficiência dos padrões educacionais no que tange prever a execução em diferentes plataformas, assim como ainda não permitir a segmentação de objetos de aprendizagem. Com isso, dois novos conjuntos de metadados foram propostos e descritos no capítulo 4 (seção 4.3), adicionando tais funcionalidades aos padrões educacionais.

Para validar a viabilidade técnica da proposta de desenvolvimento interoperável, o capítulo 5 descreve a metodologia de escolha e conversão dos objetos de aprendizagem “De onde vem a TV?” e “Outras Infâncias”, recursos esses já existentes para plataforma Web, para serem acessados de forma interoperável. Também foi implementado um OA novo, o “Viva Saudável”, já projetado de forma interoperável. Uma vez implementados, os mesmos foram documentados com as extensões de metadados propostas nessa dissertação, demonstrando também seu uso e a utilidade dos mesmos. Por fim, o capítulo 5 compara as abordagens existentes com as novas propostas deste trabalho.

Dessa forma, este trabalho atingiu os objetivos propostos, sendo que as principais contribuições desta dissertação podem ser resumidas nos seguintes itens:

- Levantamento dos principais recursos disponíveis em termos de formatos de mídia e linguagens de programação existentes nas plataformas Web, móvel e de TV digital;
- Levantamento das principais configurações possíveis de terminais de TV digital e seus cenários de uso na educação;
- Levantamento bibliográfico dos diferentes padrões de metadados educacionais, além dos disponíveis para as plataformas Web e de TV digital;
- Especificação de diretrizes para possibilitar a criação de objetos de aprendizagem interoperáveis, a qual foi adotada como recomendação pelo padrão nacional OBAA;
- Desenvolvimento de um *framework* para criação de objetos de aprendizagem interoperáveis baseados em hiperdocumento, denominado *Interop*;
- Proposta de duas extensões de metadados ao padrão LOM, as quais foram adotadas como parte integrante do conjunto de metadados do padrão nacional OBAA;
- Levantamento dos principais tipos de mídia componentes dos objetos de aprendizagem presentes em diversos repositórios brasileiros;
- Implementação de três objetos de aprendizagem para uso nas plataformas Web, móvel e de TV digital, os quais estão disponíveis e podem ser acessados a partir do Portal OBAA (OBAA, 2010) e servem de diretriz para o desenvolvimento de novos objetos de aprendizagem.

Além disso, alguns resultados obtidos ao longo deste trabalho pelo grupo de pesquisa foram publicados em artigos que permitiram divulgar e discutir as abordagens propostas nesta dissertação. Em (ROESLER et al., 2009) foi apresentada a arquitetura geral para possibilitar a criação, gerenciamento e entrega de conteúdo para diferentes plataformas. Em (BORDIGNON et al, 2009) foi apresentada uma proposta inicial de recomendações de usabilidade e de formatos de mídias de uso comum nas três plataformas, baseadas em acesso Web. Em (VARELLA et al, 2009-b) foi apresentado o *framework Interop*, utilizado para implementação dos objetos de aprendizagem. Já no âmbito do padrão OBAA, o qual adotou as contribuições desta pesquisa, foram publicados os artigos (BEZ et al, 2009), (BEZ et al, 2010) e (VICARI et al, 2010).

Conforme conclusões apresentadas, esta dissertação trouxe diversas contribuições. A partir da base apresentada neste trabalho, alguns trabalhos futuros são sugeridos, estendendo o que foi desenvolvido até o momento, como:

- **Especificação de ferramentas de apoio à geração de conteúdo interoperável:** o desenvolvimento de objetos de aprendizagem interoperáveis ainda depende de conhecimento técnico para implementação. Portanto, a especificação e criação de ferramentas de autoria que permitam aos professores criar objetos de aprendizagem por si só, mesmo que simples, é de grande valia;
- **Ampliação com base em novas tecnologias emergentes:** à medida que surgem novos formatos de mídia e linguagens de programação, essas podem se tornar opções para criação de objetos interoperáveis. Um exemplo disso é a avaliação da linguagem Java para interoperabilidade, que poderá ser realizada assim que a normatização do Gingga-J estiver definida e ambientes de testes estiverem

disponíveis para TV digital. O HTML 5 é outra tecnologia que está se popularizando e vem sendo ampliado o suporte de suas funcionalidades pelos diversos navegadores. Com isso, pode-se ampliar e gerar novas alternativas às já apresentadas neste trabalho;

- **Validação com novos terminais de acesso:** frequentemente novos celulares e *smartphones* são introduzidos no mercado e, em se tratando de TV digital, no momento só está disponível para testes a implementação de referência do GINGA-NCL, sendo que ainda nem estão disponíveis *set-top boxes* comerciais com o *middleware* GINGA embarcado. Dessa forma, à medida que novos dispositivos se tornam disponíveis e novos recursos são agregados, é importante ampliar as validações e recomendações deste trabalho;
- **Integração do *framework Interop* com as extensões de metadados:** ao invés de informar os endereços de acesso aos objetos de aprendizagem diretamente no código fonte, o *framework Interop* poderá no futuro interpretar essas informações diretamente das informações de seus metadados, centralizando as informações em um único local, evitando retrabalho de documentação e automatizando ainda mais o processo;
- **Adaptação do roteiro do objeto de aprendizagem com base no conhecimento do usuário:** além de prever a execução em diferentes plataformas, outra forma de adaptação é em relação ao perfil do usuário que está acessando o objeto de aprendizagem. Podem ser geradas novas extensões de metadados para prever tais situações, assim como acrescentar novas funcionalidades no *framework Interop* para que o mesmo realize adaptações de roteiro em função do conhecimento dos seus utilizadores;
- **Avaliação com usuários:** uma vez que esta dissertação se concentrou nas questões de viabilidade técnica, podem ser exploradas continuidades na área educacional e pedagógica. Isso pode ser feito, por exemplo, através da realização de entrevistas com usuários, buscando avaliar a usabilidade e eficiência do aprendizado a partir do acesso aos objetos de aprendizagem por meio das diferentes plataformas.

## APÊNDICE A – TERMINAIS DE TV DIGITAL E POSSÍVEIS CENÁRIOS EDUCACIONAIS

Visto que é opcional aos fabricantes que seus receptores disponibilizem um ou mais dos recursos de interatividade, como *middleware*, acesso ao canal de interatividade (acesso à internet) e armazenamento em massa, alguns recursos educacionais poderão ser disponibilizados em alguns terminais de acesso, enquanto em outros não, por não prover o suporte necessário.

Como o SBTVD não faz nenhuma classificação de perfis, exceto em relação a diferenciação *full-seg* e *one-seg*, já apresentada, considera-se ser útil uma breve descrição das diferentes combinações de recursos (neste trabalho denominadas “configurações”) que poderão estar presentes nos diversos *set-top boxes* disponíveis no mercado. Acredita-se com isso, facilitar a visualização dos possíveis cenários em que a educação pode ser utilizada sobre a plataforma de TV digital, com base nos recursos disponíveis em cada perfil de *set-top boxes*.

### A.1 Configuração 1 – Terminal de Acesso Básico

Conteúdos educacionais de áudio e vídeo são transmitidos continuamente por *broadcast*, sendo que vários conteúdos podem ser enviados simultaneamente em vários canais. Portanto, pode-se enviar um vídeo em alta definição ou então, até quatro vídeos simultaneamente em definição padrão (compondo os chamados canais virtuais).

Portanto, se pode citar como exemplo a transmissão de “campanhas de conscientização”, que se referem a conscientizações massivas sobre questões relevantes ao crescimento de um povo, como por exemplo:

- Conscientizações de saúde: dicas de prevenção de AIDS, Dengue, DST, Stress, Obesidade, Exercícios Físicos;
- Conscientizações políticas: necessidade do voto, como escolher bem seus representantes;
- Conscientizações industriais: inovação, empreendedorismo;
- Conscientizações ambientais: separação de lixo, preservação do planeta;
- Conscientizações sociais: amor ao próximo, respeito.

Neste caso, os interessados podem escolher um entre vários canais que estarão transmitindo conteúdos educacionais simultaneamente em cada horário.

Porém, não poderão gravar o programa para vê-lo posteriormente, nem interagir com os aplicativos enviados pela emissora, junto com o áudio e vídeo.

## **A.2 Configuração 2 – Terminal de Acesso com Recurso de Armazenamento em Massa**

Com o recurso de armazenamento em massa, os interessados podem programar o receptor de TV digital para gravar o conteúdo no horário que for transmitido pela emissora e assisti-lo no horário mais conveniente. Também pode ser criado um banco de conteúdo local, sendo que o modo como estas informações são armazenadas e gerenciadas é de livre escolha dos interessados.

Neste caso, além de poder escolher qual canal se quer assistir, se pode armazenar o conteúdo de outro canal para assisti-lo depois. Ou até mesmo, gravar um curso educacional que irá passar em um horário em que o interessado não possa assistir.

Sendo que, se num mesmo horário estiverem sendo transmitidos dois programas distintos, como por exemplo, um programa sobre Obesidade, e correlato a este, um programa que conscientiza sobre a prática de Exercícios Físicos mostrando como fazê-los e em que períodos é melhor praticá-los, o interessado pode assistir em tempo real o programa sobre Obesidade, e simultaneamente estar gravando o programa de conscientização da prática de Exercícios Físicos para vê-lo depois.

Outra possibilidade para este cenário é o envio de um conjunto de metadados que descreve a programação futura (por exemplo, a programação das próximas duas semanas), e o usuário seleciona quais programas o PVR deverá gravar (manualmente ou automatizado por meio de um perfil).

## **A.3 Configuração 3 – Terminal de Acesso com *Middleware* e Sem Canal de Interatividade**

Já com a possibilidade do usuário interagir com o *set-top box*, as possibilidades educacionais aumentam bastante, visto que além de assistir a programação o usuário pode realizar outras tarefas simultaneamente, como por exemplo, responder um questionário, participar de um jogo, verificar seu nível de estresse, ou seja, executar aplicações que são transmitidas juntamente com o áudio e vídeo.

Caso o usuário possua recurso de armazenamento em massa, além de assistir posteriormente um programa gravado, poderá interagir com as aplicações que foram transmitidas juntamente com o programa, pois essas também ficarão armazenadas em seu *set-top box*.

Agora, se o usuário não possuir recurso de armazenamento em massa, poderá executar a aplicação durante toda a exibição do programa, pois durante esse período, a aplicação estará sempre sendo transmitida em *broadcast*.

Exemplificando o ambiente, pode-se imaginar a transmissão de um curso de matemática, onde o usuário poderá realizar um teste durante a transmissão do programa, sendo que as respostas são validadas no próprio receptor, sem a necessidade do envio de informações a qualquer servidor. Nesse cenário, o usuário não pode interagir com o tutor (caso haja), nem participar de fóruns e discussões sobre os assuntos. Também não pode resolver dúvidas através do próprio terminal de acesso, pois, para isso, é necessário que o interessado possua um canal de interatividade.

#### **A.4 Configuração 4 – Terminal de Acesso com *Middleware* e com Canal de Interatividade**

O canal de interatividade incrementa ainda mais a gama de recursos que a TV digital pode oferecer, pois o usuário tem a possibilidade de participar ativamente da programação, interagindo não só com o *set-top box*, mas com outros usuários, bem como com quem está transmitindo o programa, por exemplo.

O conceito de “curso síncrono” permite à população acesso a eventos que estão sendo transmitidos ao vivo, como palestras, cursos, conferências nas mais diversas áreas, etc. Com o recurso do canal de interatividade, é viabilizada a ideia onde o usuário deixa de ser somente um telespectador e passa a ser um participante do programa. Visto que além de executar aplicações no seu terminal de acesso, pode enviar informações e comunicar-se com outras aplicações.

Num curso sobre direção defensiva, o usuário pode enviar perguntas ao apresentador, assim como participar de debates ao vivo, enviando sua opinião ou crítica sobre o assunto. Além disso, poderia consultar os pontos que possui em sua habilitação, enviando o número da mesma. Pois com o canal de interatividade, é possível que a aplicação transmita informações através do terminal de acesso para quaisquer servidores de dados, tornando possível sua integração com qualquer outra aplicação acessada através da internet.

#### **A.5 Configuração 5 – Terminal de Acesso com Canal de Interatividade e sem *Middleware***

Caso o aparelho do usuário não possua *middleware*, ainda assim ele pode fazer uso do canal de interatividade, pois o receptor de TV digital pode estar munido de aplicações embarcadas que não necessitem de *middleware* para serem executadas, como por exemplo, um *browser*, um cliente de e-mail, etc.

Essas aplicações deverão estar armazenadas no próprio *set-top box* e dependem do fabricante do aparelho disponibilizá-las ou não, pois não existe uma norma obrigando a suportar tais aplicativos. Se o terminal de acesso possuir esse recurso e também alguma dessas aplicações, poderá utilizar a TV digital para navegar na internet como se a mesma fosse um computador.

## APÊNDICE B – METADADOS TÉCNICOS PROPOSTOS

Tabela B.1 Conjunto de metadados propostos como extensão do padrão LOM, seção 4: *Technical*.

<i>Nr.</i>	<i>Nome</i>	<i>Descrição</i>	<i>Cardinalidade</i>	<i>Domínio</i>	<i>Tipo de Dado</i>	<i>Exemplo</i>
4.8	<i>SupportedPlatforms</i>	Lista de plataformas digitais para as quais o Objeto de Aprendizagem está previsto. Atualmente estão previstos três tipos básicos de plataformas digitais para disponibilização de OAs: Web, DTV e Mobile. Este item não é obrigatório, para manter a compatibilidade com o LOM, mas é recomendado seu preenchimento.	0..N	Mobile, DTV, Web	string	“DTV”, “Web”
4.9	<i>PlatformSpecificFeatures</i>	Conjunto de características técnicas das mídias específicas desenvolvidas para cada plataforma para a qual o Objeto de Aprendizagem foi previsto. Deverá ser criado um registro deste conjunto de metadados para cada plataforma suportada pelo OA e cujas informações técnicas diferem das informações técnicas já descritas no	0..N	-	container	-



		item 4 (Technical), ou seja, apenas quando mídias diferentes forem disponibilizadas para cada plataforma.				
4.9.1	<i>PlatformType</i>	Tipo da plataforma digital à qual se aplicam os seguintes parâmetros. Segue o mesmo vocabulário de tipos de plataforma usado no item 4.8.	1	Mobile, DTV, Web	String	"DTV", "Web"
4.9.2	<i>SpecificFormat</i>	Formato da mídia criada para utilização na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.1, porém aplicadas à mídia específica. Tamanho especificado em bytes (IEEE-LOM 4.1).	1..N	Tipos MIME baseados no registro IANA (RFC2048:1996) ou "non-digital_" (IEEE-LOM 4.1)	String (500) (IEEE-LOM 4.1)	"video/mpeg" "application/x-toolbook" "text/html" (IEEE-LOM 4.1)
4.9.3	<i>SpecificSize</i>	Tamanho da mídia (em bytes) para utilização na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.2, porém aplicadas à mídia específica.	1	ISO/IEC 646:1991, mas somente os dígitos de "0".."9" (IEEE-LOM 4.2)	String (30) (IEEE-LOM 4.2)	"4200" (IEEE-LOM 4.2)
4.9.4	<i>SpecificLocation</i>	Uma sequência de caracteres utilizada para acessar a mídia criada especialmente para utilização na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.3, porém aplicadas à mídia específica.	1	ISO/IEC 10646-1:2000 (IEEE-LOM 4.3)	String (1000) (IEEE-LOM 4.3)	"HTTP://www..."
4.9.5	<i>SpecificRequirement</i>	Capacidades técnicas necessárias na plataforma definida no item 4.9.1 para utilizar esta mídia específica. Segue as mesmas definições e regras do item 4.4, porém aplicadas à mídia específica.	1..N	ISO/IEC 10646-1:2000 (IEEE-LOM 4.4)	container	-

4.9.5.1	<i>SpecificOrComposite</i>	Agrupamento de múltiplos requisitos para a mídia na plataforma específica definida no item 4.9.1. Segue as mesmas definições e regras do item 4.4.1, porém aplicadas à plataforma específica.	1..N	(IEEE-LOM 4.4.1)	(IEEE-LOM 4.4.1)	(IEEE-LOM 4.4.1)
4.9.5.1.1	<i>SpecificType</i>	O tipo de tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.1, porém aplicadas à plataforma específica.	1	Sistema operacional, navegador, middleware	Vocabulário definido no domínio	-
4.9.5.1.2	<i>SpecificName</i>	O nome da tecnologia requerida na plataforma específica. Nota 1: O valor para este elemento de dados pode ser derivado de 4.9.2: Technical.PlatformSpecificFeatures.SpecificFormat automaticamente onde, por exemplo, “vídeo/3gp” implica múltiplos sistemas operacionais (“multi-os”). Nota 2: Este domínio inclui os valores mais comuns utilizados no momento em que este padrão foi aprovado. Novos valores poderão ser incluídos. Segue as mesmas definições e regras do item 4.4.1.2, porém aplicadas à plataforma específica.	1	Se 4.9.5.1.1:SpecificType pe=“operating system”, então: pc-dos, ms-windows, macos, unix, multi-os, none Se 4.9.5.1.1:SpecificType pe=“browser” então: any, netscape-communicator, ms-internet-explorer, opera, amaya, mozilla-Firefox, apple-safari, google-chrome Se 4.9.5.1.1:SpecificType pe=“middleware” então:	Vocabulário definido no domínio	-

				ginga, mhp, arrib, davic, dase, gem, any, none		
4.9.5.1.3	<i>SpecificMinimumVersion</i>	Versão mínima da tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.3, porém aplicadas à plataforma específica.	1	ISO/IEC 10646-1:2000 (IEEE-LOM 4.4.1.3)	String (30) (IEEE-LOM 4.4.1.3)	"4.2" (IEEE-LOM 4.4.1.3)
4.9.5.1.4	<i>SpecificMaximumVersion</i>	Maior versão aceitável da tecnologia requerida na plataforma específica. Segue as mesmas definições e regras do item 4.4.1.4, porém aplicadas à plataforma específica.	1	ISO/IEC 10646-1:2000 (IEEE-LOM 4.4.1.4)	String (30) (IEEE-LOM 4.4.1.4)	"6.2" (IEEE-LOM 4.4.1.4)
4.9.6	<i>SpecificInstallationRemarks</i>	Instruções de instalação do objeto de aprendizagem na plataforma especificada no item 4.9.1. Segue as mesmas definições e regras do item 4.5, porém aplicadas à plataforma específica.	1	-	String(1000) (IEEE-LOM 4.9.1)	("en," "Unzip the zip file and launch index.html in your web browser.") (IEEE-LOM 4.9.1)
4.9.7	<i>SpecificOtherPlatformRequirements</i>	Informações sobre outros requisitos de software e hardware necessários na plataforma definida no item 4.9.1. Segue as mesmas definições e regras do item 4.6, porém aplicadas à plataforma específica.	1	- (IEEE-LOM 4.6)	String(1000) (IEEE-LOM 4.6)	("en," "sound card"), ("en," "runtime X") (IEEE-LOM 4.6)

## APÊNDICE C – METADADOS DE SEGMENTAÇÃO PROPOSTOS

Tabela C.1: Conjunto de metadados propostos como extensão do padrão LOM a partir de metadados do padrão TV-Anytime.

<i>Nr.</i>	<i>Nome</i>	<i>Descrição</i>	<i>Cardinalidade</i>	<i>Domínio</i>	<i>Tipo de Dado</i>	<i>Exemplo</i>
11	<i>SegmentInformationTable</i>	Grupo que contém o conjunto de informações de segmentação dos objetos de aprendizagem e de grupos de segmentos dos objetos de aprendizagem.	0..1	-	container	-
11.1	<i>SegmentList</i>	Conjunto de informações de segmentos.	0..N	-	-	-
11.1.1	<i>SegmentInformation</i>	Agrupamento das informações de um segmento ( <i>SegmentInformationType</i> do TV-Anytime)	1..N	TV- Anytime	String (1000)	-
11.1.1.2	<i>SegmentIdentifier</i>	Identificador único do segmento nesse objeto de aprendizagem. É um campo alfanumérico	1	-	String(100)	1, 222, 12, a, b, exercício1, ...
11.1.1.3	<i>SegmentTitle</i>	Título do segmento. Segue as mesmas definições e regras do item LOM 1.2: <i>Title</i> , porém aplicadas ao segmento.	1	-	String (1000)	Exercícios sobre receptores de TV Digital

11.1.1.4	<i>SegmentDescription</i>	Descrição do conteúdo do segmento. Segue as mesmas definições e regras do item LOM 1.4: <i>Description</i> , porém aplicadas ao segmento.	1	-	<i>String (2000)</i>	"Seção de exercícios do módulo de introdução à TVD."
11.1.1.5	<i>SegmentKeyword</i>	Palavras-chave referentes ao segmento. Segue as mesmas definições e regras do item LOM 1.5: <i>Keyword</i> , porém aplicadas ao segmento.	1..N	-	<i>String(1000)</i>	"TV Digital, receptores, exercícios"
11.1.1.6	<i>SegmentMediaType</i>	Descreve se é um documento texto ( <i>document</i> ), hiperdocumento ( <i>hyperdocument</i> ), arquivo multimídia ( <i>audio</i> ou <i>video</i> ), ou outros ( <i>other</i> ).	1	<i>Document, Hyperdocument, audio, video, image, other</i>	<i>Vocabulário descrito no domínio</i>	<i>video hyperdocument</i>
11.1.1.7	<i>Start</i>	Início do segmento. Se o segmento for originado de um arquivo multimídia, deverá indicar o tempo de início (MPEG-7 <i>MediaTimeType</i> ). Se for um documento texto, indica a página e, opcionalmente, a linha inicial. Se for um hiperdocumento, a página, seção de uma página ou mídia inclusa no hiperdocumento. No caso de imagens, a linha e coluna de início do segmento.	1	-	<i>String</i>	"PT00H05M" "módulo1/receptores.xhtml#exercicios" "Page20Line10"
11.1.1.8	<i>End</i>	Fim do segmento. Se segmento for originado de um arquivo multimídia, deverá indicar o tempo de fim (MPEG-7 <i>MediaTimeType</i> ). Se for um documento texto, indica a página e, opcionalmente, a linha final. Se for um hiperdocumento, este metadado não será utilizado.	1	-	<i>String</i>	"PT00H05M" "Page20Line10"

11.2	<i>SegmentGroupList</i>	Conjunto dos grupos de segmento	0..1	-	<i>Container</i>	-
11.2.1	<i>SegmentGroupInformation</i>	Conjunto de informações do grupo de segmentos (SegmentGroupInformationType do TV-Anytime).	1..N	-	<i>Container</i>	-
11.2.1.1	<i>Identifier</i>	Identificador do grupo de segmento. Deve ser único no objeto de aprendizagem.	1	-	<i>String</i>	<i>1, 23,1000, Grupo1, Exercícios</i>
11.2.1.2	<i>GroupType</i>	Tipo de agrupamento. Seguirá a mesma sintaxe de TVA: SegmentGroupTypeType	1	<i>highlights, bookmarks, themeGroup, preview, activities, other</i>	<i>Vocabulário definido no domínio</i>	<i>Highlights Bookmarks</i>
11.2.1.3	<i>Title</i>	Título do segmento. Segue as mesmas definições e regras do item LOM 1.2: Title, porém aplicadas ao grupo de segmentos.	1	-	<i>String(1000)</i>	<i>Atividades e Exercícios do OA.</i>
11.2.1.4	<i>Description</i>	Descrição do conteúdo do segmento. Segue as mesmas definições e regras do item LOM 1.4: Description, porém aplicadas ao grupo de segmentos.	1	-	<i>String (2000)</i>	<i>Grupo contendo os segmentos com os exercícios deste objeto de aprendizagem.</i>
11.2.1.5	<i>Keyword</i>	Palavras-chave referentes ao segmento. Seguem as mesmas definições e regras do item LOM 1.5: Keyword, porém aplicadas ao grupo de segmentos.	1..N	-	<i>String</i>	<i>Exercícios, Atividades</i>
11.2.1.6	<i>Segments</i>	Segmentos que fazem parte do grupo	1	-	<i>Container</i>	-
11.2.1.6.1	<i>Identifier</i>	Código identificador único do segmento (11.1.1.2:Identifier) que pertence a este grupo	1..N	-	<i>String</i>	<i>"1", "222", "12", "a", "b", "exercício1"</i>

## APÊNDICE D – FERRAMENTAS UTILIZADAS

Para realização do presente trabalho, algumas ferramentas já desenvolvidas e disponíveis na Internet foram utilizadas. Algumas ferramentas se referem à conversão de arquivos do formato original para formatos interoperáveis entre as plataformas. Outras ferramentas são as que irão compor o servidor de aplicação, responsável por: (a) transmitir *streaming* de vídeo e (b) distribuir os aplicativos com as adaptações necessárias para cada plataforma.

Para escolha das ferramentas, inicialmente realizou-se um estudo comparativo entre as principais ferramentas, o qual será apresentado nas próximas seções.

### D.1 Ferramentas para Conversão de Formatos

Duas ferramentas foram analisadas para realização da conversão de áudio e vídeo, por suportar uma grande variedade de formatos e serem disponibilizados de forma gratuita: o SUPER (*Simplified Universal Player Encoder & Renderer*), versão *v2009.build.35*, e o VLC Media Player, versão 0.9.8a.

O SUPER (ERIGHTSOFT, 2010), ilustrado pela Figura 47, é uma interface gráfica disponível para o sistema operacional Windows que utiliza diversos *players* e codificadores gratuitos para realizar conversão entre formatos de áudio e vídeo, dentre eles o FFmpeg<sup>21</sup>, MEncoder, MPlayer<sup>22</sup>, x264<sup>23</sup>, FFmpeg2theora<sup>24</sup> e theora/vorbis RealProducer plugIn<sup>25</sup>. Além de ser um *software* gratuito, devido ao grande número de *decoders* que o mesmo utiliza, oferece suporte para conversão entre praticamente quaisquer formatos.

---

<sup>21</sup> Disponível em: <<http://ffmpeg.org/>>. Acesso em: mar. 2010.

<sup>22</sup> Disponível em: <<http://www.mplayerhq.hu/>>. Acesso em: mar. 2010.

<sup>23</sup> Disponível em: <<http://www.videolan.org/developers/x264.html>>. Acesso em: mar. 2010.

<sup>24</sup> Disponível em: <<http://v2v.cc/~j/ffmpeg2theora/>>. Acesso em: mar. 2010.

<sup>25</sup> Disponível em: <<http://theora.org/>>. Acesso em: mar. 2010.

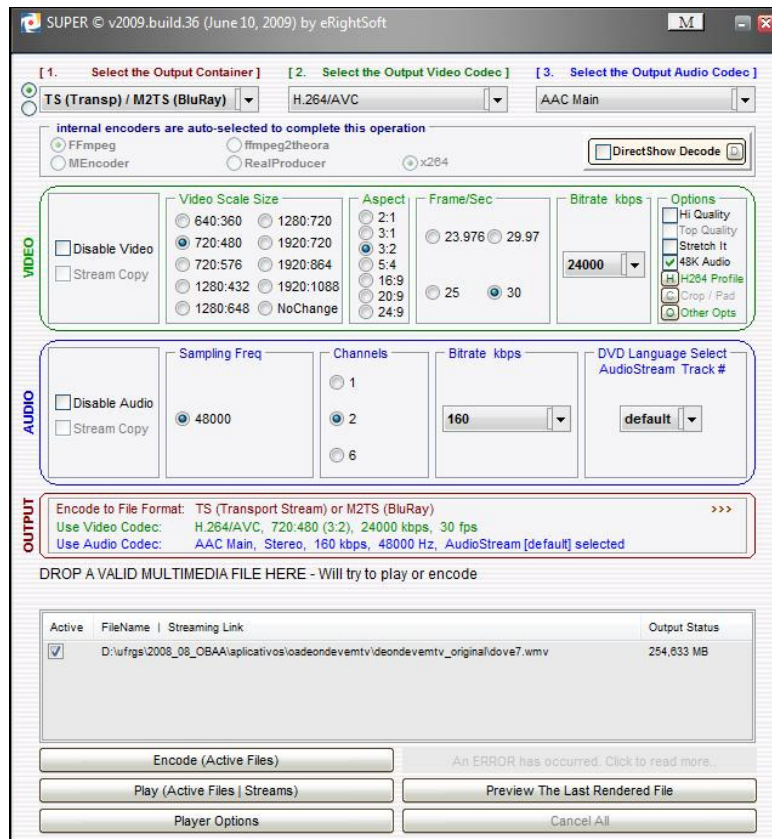


Figura 47. Ferramenta SUPER, utilizada de conversão de formatos de áudio e vídeo (ERIGHTSOFT, 2010).

Sua utilização prossegue da seguinte maneira: o usuário escolhe os arquivos que serão convertidos ou executados, então escolhe o formato de codificação de vídeo, de áudio, além de outros detalhes conforme ilustrado na Figura 47 e que correspondem ao formato de saída dos arquivos selecionados. Na sequência inicia o processo de conversão/execução através dos botões *encode* ou *play*.

Em sua utilização, não se sabe qual codificador/decodificador está sendo utilizado especificamente para cada formato, dentre o conjunto de bibliotecas que o SUPER reúne, pois ele seleciona automaticamente um codificador para executar de acordo com o formato escolhido. O usuário só precisa especificar os formatos do áudio/vídeo e o SUPER resolve automaticamente a aplicação desses formatos.

Já o VLC Media Player (VIDEOLAN, 2010), ilustrado pela Figura 48, é um *player* multimídia de código aberto que possui suporte a vários formatos de vídeo e áudio. Além disso, tem suporte a vários protocolos de *streaming*, podendo ser usado como servidor de vídeo, com transmissão em *unicast* ou *multicast*. Oferece também suporte a vários protocolos como UDP, HTTP, RTP, RTPS, entre outros. O VLC tem versões para vários sistemas operacionais, tais como Windows, Mac OS, FreeBSD, GNU/Linux, BeOS, etc.



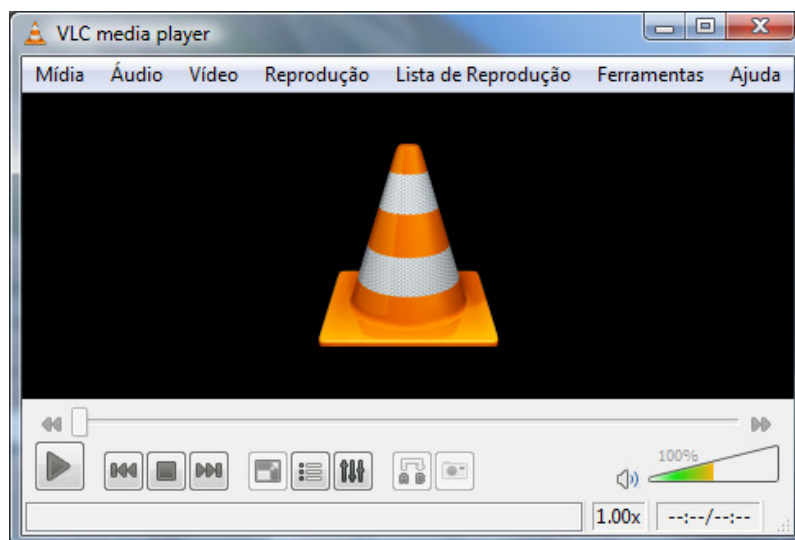


Figura 48. Interface do VLC Media Player.

O VLC, por meio da opção “Mídia → Converter/Salvar” (Figura 49), pode ser utilizado também como ferramenta de conversão de mídias.

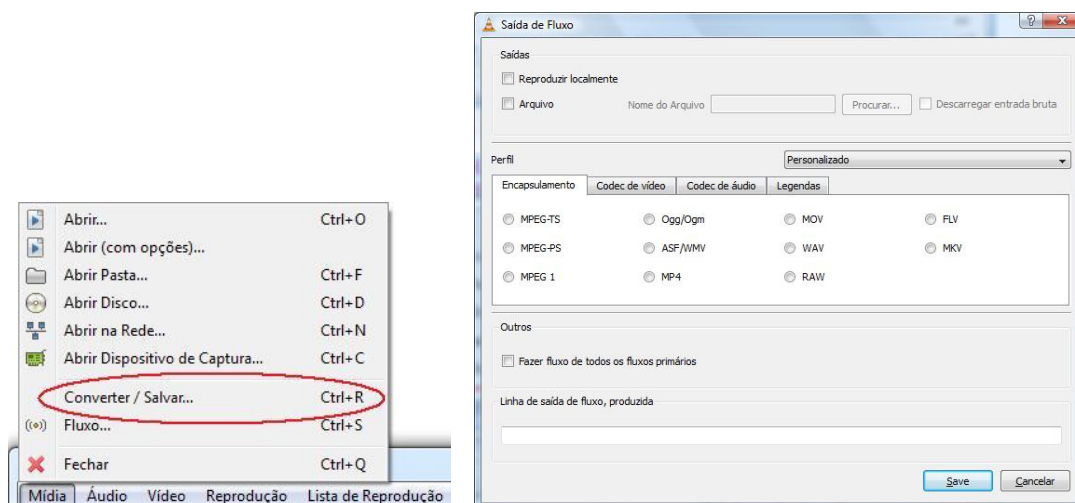


Figura 49. Conversão de mídias no VLC Media Player.

Tendo analisado as ferramentas disponíveis, neste trabalho optou-se pela ferramenta SUPER para conversão das mídias para os formatos requeridos. Essa escolha se deve ao fato do mesmo dar suporte à conversão para todos os formatos necessários, enquanto que o VLC não possui suporte ao formato de arquivo 3gp, que é o formato padrão utilizado em dispositivos móveis.

Porém, não está descartado que, caso necessário, o VLC seja utilizado no futuro, pois ele possui algumas opções não existentes no SUPER, a citar: a função de transmissão de vídeo via *Transport Stream* (forma de transmissão para TV digital) e versões para sistemas operacionais diferentes do *Windows*.

## D.2 Ferramentas para Compor o Servidor de Aplicação

Um servidor de aplicação pode ser entendido como um conjunto de aplicações que são executadas em uma máquina hospedeira para dar-lhe alguma funcionalidade específica. Essas aplicações tornam essa máquina capaz de, por exemplo, armazenar

*web sites* e de receber e processar requisições Web (HTTP) ou transferir vídeos. Conforme será visto a seguir, foram exatamente essas duas funcionalidades utilizadas por esta pesquisa.

### D.2.1 Ferramentas de Streaming

A primeira ferramenta de *streaming* estudada foi o Helix DNA Server, que é um servidor de vídeo sob demanda, ou seja, a partir da requisição de um cliente, é possível iniciar a transmissão de uma *stream* de áudio/vídeo. Além disso, pode-se transmitir em diversos protocolos, como RTP, RTSP, HTTP, UDP e TCP.

O Helix DNA Server é a versão *open source* do Helix Server, servidor de *streaming* proprietário da Real Networks (REAL NETWORKS, 2010). Contudo, os formatos suportados por esse servidor, na versão gratuita, não atendem as recomendações feitas pelo grupo, visto que o Helix apresenta suporte apenas à transmissão de arquivos MP3 Áudio (.mp3) e RealAudio, RealVideo (.rm, .ra, .rv). Conquanto, ele é gratuito e serve como apoio. Nos nossos testes, transmitir conteúdo audiovisual via *stream* mostrou-se mais eficiente do que realizar o *download* do vídeo. Além disso, esse método proporciona uma melhor experiência ao usuário.

A partir daí, testou-se a versão paga do Helix (Helix Server v12), que oferece então, suporte aos formatos MP4 e 3GP, obtendo-se então, resultados positivos. Ou seja, adquirindo-se a ferramenta é possível realizar a transmissão via *streaming* dentro dos padrões propostos.

Outra ferramenta estudada foi o Darwin Streaming Server (APPLE, 2010-a), que é a versão *open source* do Apple's QuickTime Streaming Server e possui suporte a transmissão através dos protocolos RTP e RTSP, bem como aos formatos de arquivo segundo padrões 3GPP e MPEG-4, .3gp e .mp4, respectivamente. Além de ser gratuita, sua instalação é extremamente simples e a ferramenta possui uma interface *web* para configuração.

O VLC (VIDEOLAN, 2010), além de atuar como um conversor de formatos de áudio e vídeo (conforme descrito anteriormente) é também capaz de atuar como servidor de *stream* de áudio e vídeo em modo *multicast*, ou seja, pode-se transmitir um vídeo para toda uma rede, simulando uma emissora de televisão.

Outro ponto positivo é o fato do VLC transmitir em formato TS (Transport Stream), o qual o torna muito similar às transmissões de TV, embora não gere as tabelas SI/PSI e demais informações definidas no sistema brasileiro (como dados).

Porém, apesar de estudado o servidor VLC, o mesmo ainda não foi utilizado, pois o Gíngua-NCL *Virtual Set-top Box* executa os arquivos de vídeo diretamente de um diretório local, previamente transferido. Mas a partir desse estudo, não está descartada a possibilidade futura de utilizar o VLC como ferramenta base para simulação de um multiplexador, permitindo simular transmissões em um ambiente ainda mais próximo ao real.

Dentre as ferramentas analisadas, optou-se em utilizar em seu ambiente de testes o *Darwin Streaming Server*, pois o mesmo é capaz de fornecer sob demanda, *streaming* de áudio/vídeo nos formatos mp4 e 3gp, nas codificações propostas para o padrão. Além disso, utiliza o protocolo RTSP, que é um padrão aberto e suportado tanto na Web quanto em dispositivos móveis.

Essa ferramenta é de fácil configuração e possui uma interface de administração Web que permite ajustar as configurações do servidor remotamente. Essa interface pode ser acessada de qualquer *web browser* através da url do servidor, por exemplo:

*http://hostname:1220*

Onde *hostname* é o nome ou endereço IP do servidor de *streaming* e 1220 é o número da porta.

A Figura 50 ilustra a página do administrador do servidor de *streaming*. Nessa página deve-se informar o caminho do diretório no servidor que conterá as mídias a serem transmitidas (campo “*Media Directory*”). Desse modo, qualquer arquivo de mídia, compatível com o DSS, que for colocado nessa pasta poderá ser acessado via *streaming*.

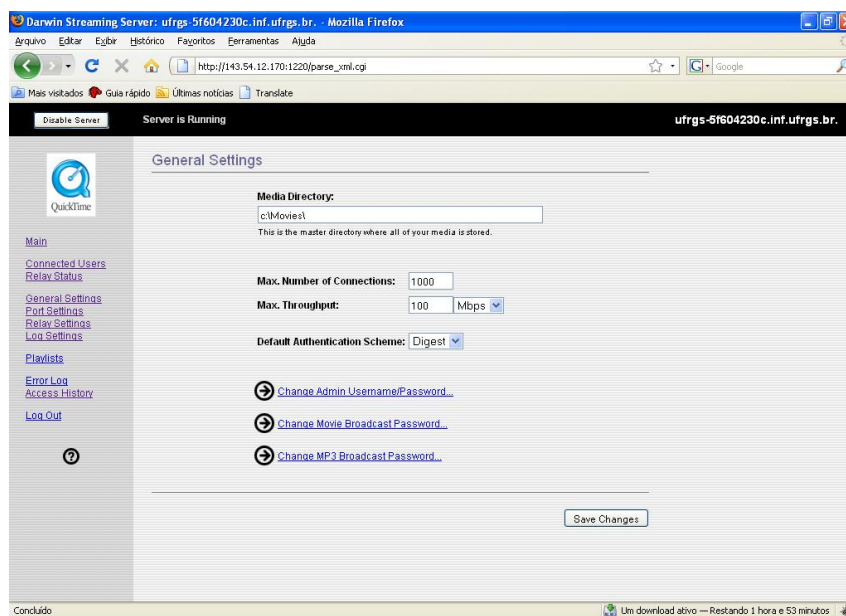


Figura 50. Página de configuração do DSS.

Os vídeos disponíveis no servidor poderão ser acessados de qualquer *player* de *streaming* compatível através da seguinte URL:

*rtsp://hostname/video.mp4*

Onde *hostname* é o nome ou endereço IP do servidor de *streaming* e *video.mp4* é o nome do arquivo de mídia a ser acessado.

Antes que os vídeos possam ser transmitidos, é necessário prepará-los para que o *streaming* possa ser realizado. O servidor de *streaming* requer que as mídias *.mov* *.mp4* *.3gp* possuam “*hint traks*”. As *hint tracks* informam ao servidor exatamente como transmitir a mídia. Elas podem ser adicionadas ao vídeo através de ferramentas como Quick Time Pro<sup>26</sup> ou Mp4Box<sup>27</sup>.

<sup>26</sup> Disponível em: <<http://www.apple.com/quicktime/pro/>>. Acesso em: mar. 2010.

<sup>27</sup> Disponível em: <<http://www.videohelp.com/tools/mp4box>>. Acesso em: mar. 2010.

### D.2.2 Servidor Web

Para o desenvolvimento da aplicação foi escolhida a tecnologia JavaEE<sup>28</sup>, um dos *frameworks* mais populares para desenvolvimento Web atualmente. Tratando-se de ambiente Web, foram utilizadas também JSP 2.0<sup>29</sup> e JSF 1.2<sup>30</sup> (*Java Server Pages* e *Java Server Faces*, respectivamente), pois essas facilitam e agilizam o desenvolvimento desse tipo de aplicação.

O Glassfish versão 2ur2 (SUN MICROSYSTEMS, 2010-h) foi escolhido como servidor de aplicação Web, pois, além de fornecer suporte às tecnologias citadas, é gratuito, robusto e amplamente utilizado atualmente.

---

<sup>28</sup> Disponível em: <<http://java.sun.com/javaee/>>. Acesso em: ago. 2010.

<sup>29</sup> Disponível em: <<http://java.sun.com/products/jsp/>>. Acesso em: ago. 2010.

<sup>30</sup> Disponível em: <<http://java.sun.com/javaee/javaserverfaces/>>. Acesso em: ago. 2010.

## REFERÊNCIAS

3GPP. The 3rd Generation Partnership Project. 3GPP Specification series. 2010-a. Disponível em: <<http://www.3gpp.org/ftp/Specs/html-info/26-series.htm>>. Acesso em: ago. 2010.

3GPP. The 3rd Generation Partnership Project 2010-b. Disponível em: <<http://www.3gpp.org>>. Acesso em: ago. 2010.

3GPP. Technical Specification Group, Service & Systems Aspects. Working Group 4. SA4 - Codec. 2010-c. Disponível em: <<http://www.3gpp.org/SA4>>. Acesso em: ago. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15602-1. Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação. Parte 1: Codificação de vídeo. 2008-a. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15602-1\\_2007Vc\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15602-1_2007Vc_2008.pdf)>. Acesso em: jul. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15602-2. Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação. Parte 2: Codificação de áudio. 2008-b. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15602-2\\_2007Vc\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15602-2_2007Vc_2008.pdf)>. Acesso em: jul. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15602-3. Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação. Parte 3: Sistemas de multiplexação de sinais. 2008-c. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15602-3\\_2007Vc\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15602-3_2007Vc_2008.pdf)>.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15603-1. Televisão digital terrestre – Multiplexação e serviços de informação (SI). Parte 1: SI do sistema de radiodifusão. 2008-d. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15603\\_2D1\\_2007Vc2\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15603_2D1_2007Vc2_2008.pdf)>. Acesso em: jan. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15603-2. Televisão digital terrestre – Multiplexação e serviços de informação (SI). Parte 2: Estrutura de dados e definições da informação básica de SI. 2008-e. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15603\\_2D2\\_2007Vc3\\_2009.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15603_2D2_2007Vc3_2009.pdf)>. Acesso em: jan. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15604. Televisão digital terrestre – Receptores. 2008-f. Disponível em:

<[http://www.dtv.org.br/download/pt-br/ABNTNBR15604\\_2007Vc\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15604_2007Vc_2008.pdf)>. Acesso em: jul. 2010.

ABNT – Associação Brasileira de Normas Técnicas. ABNT NBR 15606-2. Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. 2008-g. Disponível em: <[http://www.dtv.org.br/download/pt-br/ABNTNBR15606\\_2D2\\_2007Vc3\\_2008.pdf](http://www.dtv.org.br/download/pt-br/ABNTNBR15606_2D2_2007Vc3_2008.pdf)>. Acesso: julho de 2010.

ADOBE. Open Screen Project. 2009. Disponível em: <<http://www.openscreenproject.org/>>. Acesso em: jul. 2010.

ADOBE. Adobe Flash Player. 2010-a. Disponível em: <<http://www.adobe.com/br/products/flashplayer/>>. Acesso em: jul. 2010.

ADOBE. Developing Flash Lite 2.x and 3.0 applications. 2010-b. Disponível em: <<http://www.adobe.com/support/documentation/en/flashlite/>>. Acesso em: nov. 2009.

ANDREATA, J. A. InteraTV: Um Portal para Aplicações Colaborativas em TV Digital Interativa Utilizando a Plataforma MHP. Dissertação de Mestrado. Programa de Pós-graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina. 2006.

ADL. SCORM 2004 4th Edition Version 1.1 Overview. 2004. Disponível em: <<http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Overview.aspx>>. Acesso em: jan. 2010.

APPLE. Open Source Streaming Server. 2010-a. Disponível em: <<http://www.apple.com/br/quicktime/streamingserver/>>. Acesso em: ago. 2010.

APPLE. QuickTime 7. 2010-b. Disponível em: <<http://www.apple.com/quicktime/download/>>. Acesso em: jul. 2010.

BARBOSA, M. L. K.; ROESLER, V; REATEGUI, E. B.. Uma Proposta de Modelo de Interface Interoperável para Web, TV Digital e Dispositivos Móveis. RENOUE. Revista Novas Tecnologias na Educação, v. 7, p. 1-10, 2009. Disponível em: <[http://www.cinted.ufrgs.br/renote/jul2009/artigos/9d\\_maria.pdf](http://www.cinted.ufrgs.br/renote/jul2009/artigos/9d_maria.pdf)>. Acesso em: jan. 2010.

BBC. Interactive Television Design: Designing for Interactive Television v1.0 – BBCi & Interactive TV Programmes. 2006. Disponível em: <[www.bbc.co.uk/guidelines/futuremedia/desed/itv/itv\\_design\\_v1\\_2006.pdf](http://www.bbc.co.uk/guidelines/futuremedia/desed/itv/itv_design_v1_2006.pdf)>. Acesso em: jul. 2010.

BECKER, V.; FORNARI, A.; HERWEG FILHO, G. H; MONTEZ, C. Recomendações de Usabilidade para TV Digital Interativa. In: II WTVD, 2006, Curitiba. Anais do WTVD 2006 - Workshop de TV Digital, 2006. p. 27-38.

BEZ, M.R. ; SILVA, J.M.C. ; SANTOS, E. ; PRIMO, T. ; BORDIGNON, A. . OBAA project: An approach to interoperable learning objects based on Web and digital television. In: WCCE 2009 - IFIP World Conference on Computer in Education, 2009-a, Bento Gonçalves.

BEZ, M.R. ; SILVA, J.M.C. ; SANTOS, E. ; PRIMO, T. ; BORDIGNON, A. Projeto OBAA: Uma abordagem com objetos de aprendizagem interoperáveis baseados na web e na televisão digital. *Informática na Educação*, v. 12, p. 119-126, 2009-b.

BORDIGNON, A. et al. Mechanisms for interoperable content production among Web, Digital TV and Mobiles. In *Proceedings of WCCE 2009: IX IFIP WORLD CONFERENCE ON COMPUTERS IN EDUCATION*. 2009. Bento Gonçalves, Brazil.

BRASIL. Portal do Software Público Brasileiro. Comunidade Amadeus. 2010. Disponível em: <[http://www.softwarepublico.gov.br/ver-comunidade?community\\_id=9677539](http://www.softwarepublico.gov.br/ver-comunidade?community_id=9677539)>. Acesso em: jul. 2010.

CAGENIUS, T. et al. Evolving the TV experience: Anytime, anywhere, any device. *Ericsson Review*. no. 03, 2006.

CANCORE. Página oficial. 2006. Disponível em: <<http://cancore.athabascau.ca/en/index.html>>. Acesso em: ago. 2010.

CHALMERS, P. Web Usability course. Benefit from IT. Disponível em: <[http://www.pcba10167.pwp.blueyonder.co.uk/index\\_wdu01.htm](http://www.pcba10167.pwp.blueyonder.co.uk/index_wdu01.htm)>. Acesso em: jul. 2010.

CHANG, S; SIKORA, T; PURI, A. Overview of the MPEG-7 Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. v. 11, n 6, p.688-695, 2001.

COMPUSERVE. Graphics Interchange Format Version 89a. Columbus. Ohio. 1990. Disponível em: <<http://www.w3.org/Graphics/GIF/spec-gif89a.txt>>. Acesso em: jul. 2010.

CPqD. Projeto Brasileiro de Televisão Digital - Especificação Técnica de Referência. OS 40544. 2006.

DAVIC. Digital Audio Visual Council. 2009. Disponível em: <<http://www.davic.org>>. Acesso em: mai. 2010.

DCMI. Dublin Core Metadata Element Set, Version 1.1. 2009-a. Disponível em: <<http://dublincore.org/documents/dces/>>. Acesso em: jan. 2010.

DCMI. DCMI Type Vocabulary. 2009-b. Disponível em: <<http://dublincore.org/documents/dcmi-type-vocabulary/>>. Acesso: jan. 2010.

DURAND, G.; KAZAI, G.; LALMAS, M. A metadata model supporting scalable interactive TV services. In: *Proceedings of the 11th International Multimedia Modeling Conference (MMM 2005)*.

DVB. Introduction to MHP & GEM. 2008. Disponível em: <<http://www.mhp.org/introduction.htm>>. Acesso em: jul. 2010.

ECMA – Standard ECMA-262, “ECMAScript Language Specification – Edition 5”. 2009. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>>. Acesso em: julho de 2010.

ERIGHTSOFT. SUPER – Simplified Universal Player Encoder & Renderer. 2010. Disponível em: <<http://www.erightssoft.com/SUPER.html>>. Acesso em: ago. 2010.

FERNANDES, J.; LEMOS, G.; ELIAS, G. Introdução a Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas. In Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação – JAI-SBC. Anais do JAI-SBC. Salvador – BA, 2004.

FORAKER DESIGN. Usability First: Website Design. 2010. Disponível em: <<http://www.usabilityfirst.com/about-usability/website-design/>>. Acesso em: nov. 2009.

GRUPO TELEFÔNICA DO BRASIL. A sociedade da informação no Brasil: presente e perspectivas. São Paulo: Grupo Telefônica no Brasil, 2002. 241 p. CD-ROM.

HAVi. Home Audio Video Interoperability. 2009. Disponível em: <<http://www.havi.org/>>. Acesso em: mai. 2010.

IEEE. IEEE Std 1484.12.1™-2002. IEEE Standard for Learning Object Metadata. 2002.

ITEC. Information Technology — MPEG-21 Multimedia Framework. Disponível em: <[http://mpeg-21.itec.uni-klu.ac.at/cocoon/mpeg21/\\_mpeg21Parts.html](http://mpeg-21.itec.uni-klu.ac.at/cocoon/mpeg21/_mpeg21Parts.html)>. Acesso em: jan. 2010.

ITU. H.263: Video coding for low bit rate communication. 2008. Disponível em: <<http://www.itu.int/rec/T-REC-H.263/en>>. Acesso em: jul. 2010.

JISC CETIS. Centre for Educational technology & interoperability standards. 2010. Disponível em: <<http://jisc.cetis.ac.uk/>>. Acesso em: ago. 2010.

KALLIOJA, Mika. Media Asset Management Approach to Metadata in Television Production. Master’s thesis submitted for inspection for the degree of Master of Science in Technology. Finlândia. 2006.

KUNZE, J.; BAKER, T. The Dublin Core Metadata Element Set: RFC 5013. California: IETF, 2007. Disponível em: <<http://www.ietf.org/rfc/rfc5013.txt>>. Acesso em: mar. 2010.

LEITE, L.E. TV Digital Interativa: Desenvolvimento de Aplicações Interativas. In I Fórum Amazônico de TV Digital. Manaus, 2005.

LG ELECTRONICS. LG Mobile Developer Network, 2010. Disponível em: <<http://developer.lgmobile.com/>>. Acesso em: jul. 2010.

LIANG et al. Fusion of digital television, broadband Internet and mobile communications. In Wiley International Journal of Satellite Communications and Networking. Vol. 25. Issue 4. p. 409-440. 2007.



LUGMAYR, A.; NIIRANEN, S.; KALLI, S. Digital Interactive TV and Metadata: Future Broadcast Multimedia. Springer. 2004.

MARKET SHARE. Browser Market Share. 2010. Disponível em: <<http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0>>. Acesso em: ago. 2010.

MICROSOFT. Windows Media Player 11. 2010. Disponível em: <<http://www.microsoft.com/windows/windowsmedia/br/>>. Acesso em: jul. 2010.

MONTEZ, C; BECKER, V. TV Digital Interativa: conceitos, desafios e perspectivas para o Brasil. Florianópolis: Ed. da UFSC, 2005. 2ª edição.

MOTOROLA. MOTODEV, Documentation & Tools. 2010. Disponível em: <<https://developer.motorola.com/docstools/specsheets/>>. Acesso em: jul. 2010.

MOZILLA. Mozilla Developer Center. SVG in Firefox. 2009. Disponível em <[https://developer.mozilla.org/en/SVG\\_in\\_Firefox](https://developer.mozilla.org/en/SVG_in_Firefox)>. Acesso em: nov. 2009.

MOZILLA. Add-ons for Firefox. 2010. Disponível em: <<https://addons.mozilla.org/pt-BR/firefox/browse/type:7>>. Acesso em: jul. 2010.

MPEG-7. MPEG-7 Overview (ISO/IEC JTC1/SC29/WG11 N6828). 2004. Disponível em: <<http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>>. Acesso em: jan. 2010.

NOKIA. Nokia Web Browser Design Guide. Version 1.0. 2007. Disponível em: <[http://www.forum.nokia.com/info/sw.nokia.com/id/7845f71c-0c0c-400f-8c66-69b13eafa2cb/Nokia\\_Web\\_Browser\\_Design\\_Guide.html](http://www.forum.nokia.com/info/sw.nokia.com/id/7845f71c-0c0c-400f-8c66-69b13eafa2cb/Nokia_Web_Browser_Design_Guide.html)>. Acesso em: ago. 2010.

NOKIA. Device Specifications. 2010-a. Disponível em: <<http://www.forum.nokia.com/devices/>>. Acesso em: ago. 2010.

NOKIA. Design and User Experience Library v1.7. Multimedia. Mobile Video and Streaming. Video Encoding Guidelines. Video Coding. 2010-b. Disponível em: <[http://library.forum.nokia.com/topic/Design\\_and\\_User\\_Experience\\_Library/GUID-F89F765D-4FEF-4D2D-8B7D-1A9AF419B228.html](http://library.forum.nokia.com/topic/Design_and_User_Experience_Library/GUID-F89F765D-4FEF-4D2D-8B7D-1A9AF419B228.html)>. Acesso em: jul. 2010.

NOKIA. Design and User Experience Library v1.7. Multimedia. Mobile Video and Streaming. Video Encoding Guidelines. Audio Coding. 2010-c. Disponível em: <[http://library.forum.nokia.com/index.jsp?topic=/Design\\_and\\_User\\_Experience\\_Library/GUID-863510AB-F6EA-40BB-9C2A-37445CCDC9C3.html](http://library.forum.nokia.com/index.jsp?topic=/Design_and_User_Experience_Library/GUID-863510AB-F6EA-40BB-9C2A-37445CCDC9C3.html)>. Acesso em: jul. 2010.

OBAA. Portal OBAA. 2010. Disponível em: <<http://www.portalobaa.org/>>. Acesso em: ago. 2010.

OMA. ECMAScript Mobile Profile: A Wireless Markup Scripting Language. Version 1.0. 2006. Disponível em: <[http://www.openmobilealliance.org/technical/release\\_program/docs/Browsing/V2\\_2-20061020-A/OMA-WAP-ESMP-V1\\_0-20061020-A.pdf](http://www.openmobilealliance.org/technical/release_program/docs/Browsing/V2_2-20061020-A/OMA-WAP-ESMP-V1_0-20061020-A.pdf)>. Acesso em: ago. 2010.

- OMA. Wireless CSS Specification. Candidate Version 1.2. 2008. Disponível em: <[http://www.openmobilealliance.org/Technical/release\\_program/docs/Browsing/V2\\_4-20080923-C/OMA-TS-WCSS-V1\\_2-20080923-C.pdf](http://www.openmobilealliance.org/Technical/release_program/docs/Browsing/V2_4-20080923-C/OMA-TS-WCSS-V1_2-20080923-C.pdf)>. Acesso em: ago. 2010.
- PAULSON, L.D. TV comes to the mobile phone. In IEEE Computer, April 2006, v. 39, Issue 4, pp 13- 16.
- PFEIFFER, S.; SRINIVASAN, U. 2000. TV anytime as an application scenario for MPEG-7. In: Proceedings of the 2000 ACM Workshops on Multimedia. MULTIMEDIA '00. ACM, New York, NY, 89-92.
- PROJETO AMADEUS. Página Oficial do Projeto Amadeus. 2010. Disponível em: <<http://amadeus.cin.ufpe.br/index.html>>. Acesso em: jul. 2010.
- REAL NETWORKS. Products and Services. 2010. Disponível em: <<http://www.realnetworks.com>>. Acesso em: jul. 2010.
- ROELOFS, G. Portable Network Graphics. An Open, Extensible Image Format with Lossless Compression. 2010. Disponível em: <<http://www.libpng.org/pub/png/>>. Acesso em: jul. 2010.
- ROESLER, V.; BARBOSA, M.L.; VARELLA, F. A.; BORDIGNON, A. Uma Proposta de Arquitetura Interoperável entre Web, TV Digital e Dispositivos Móveis. In: SBIE 2009. XX Simpósio Brasileiro de Informática na Educação. Florianópolis, SC. 2009.
- SAMSUNG. Samsung Mobile, Device Specifications, 2010. Disponível em: <<http://developers.samsungmobile.com>>. Acesso em: jul. 2010.
- SMTPE – Society for Motion Picture and Television Engineers. SMPTE 336M-2001. Data encoding protocol using key-length-value (KLV). SMPTE Standard. 2001.
- SMTPE. Society for Motion Picture and Television Engineers. 2010. Disponível em: <[www.smpte.org](http://www.smpte.org)>. Acesso em: jan. 2010.
- SOARES, L.F.G. Programando em NCL: desenvolvimento de aplicações para middleware Ginga, TV digital e Web. Rio de Janeiro: Elsevier, 2009.
- SONY ERICSSON. Phone Specifications. 2010. Disponível em: <<https://developer.sonyericsson.com>>. Acesso em: jul. 2010.
- SUN MICROSYSTEMS. Sun Developer Network (SDN). Java SE at a Glance. 2010-a. Disponível em: <<http://java.sun.com/javase/>>. Acesso em: jul. 2010.
- SUN MICROSYSTEMS. Sun Developer Network (SDN). Java EE at a Glance. 2010-b. Disponível em: <<http://java.sun.com/javaee/>>. Acesso em: jul. 2010.
- SUN MICROSYSTEMS. Sun Developer Network (SDN). Java ME at a Glance. 2010-c. Disponível em: <<http://java.sun.com/javame/>>. Acesso em: jul. 2010.
- SUN MICROSYSTEMS. JavaFX. 2010-d. Disponível em: <<http://www.sun.com/software/javafx/>>. Acesso em: jul. 2010.

SUN MICROSYSTEMS. Sun Developer Network (SDN). Java ME. Mobile Information Device Profile (MIDP); JSR 118. 2010-e. Disponível em: <<http://java.sun.com/products/midp/>>. Acesso em: jul. 2010.

SUN MICROSYSTEMS. Sun Developer Network (SDN). Java ME. Mobile Media API (MMAPI); JSR 135. 2010-f. Disponível em: <<http://java.sun.com/products/mmapi/>>. Acesso em: jul. 2010.

SUN MICROSYSTEMS. Sun Developer Network (SDN). Applets. Code Samples and Apps. 2010-g. Disponível em: <<http://java.sun.com/applets/>>. Acesso em: jul. 2010.

SUN MICROSYSTEMS. Glassfish Community. 2010-h. Disponível em: <<https://glassfish.dev.java.net/>>. Acesso em: ago. 2010.

TAVARES, Walkyria M. Leitão. Implantação da Televisão Digital no Brasil. Brasília – DF: Câmara dos Deputados, 2001.

THE INTERNET DIGEST. Web-Safe Fonts for Your Site. 2003. Disponível em: <<http://www.theinternetdigest.net/archive/websafefonts.html>>. Acesso em: nov. 2009.

TVA. TV-Anytime Forum. S1 – Phase 1 Benchmark Features (informative) v1.1. TV-Anytime Specification, 2001.

TVA. TV-Anytime Forum. S2 system description (informative with mandatory appendix b) v1.3. TV-Anytime Specification, 2003-a.

TVA. TV-Anytime Forum. S3 metadata (normative) v1.2. TV-Anytime Specification, 2003-b.

VARELLA, F. A. Um framework para o desenvolvimento de conteúdo interoperável entre dispositivos móveis, TV Digital e Web. 2009-a. 105f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

VARELLA, F. A.; BORDIGNON, A.; ROESLER, V.; BARBOSA, M.L. Interop: um Framework para o Desenvolvimento de Conteúdo Web Interoperável entre TV Digital, Dispositivos Móveis e Computadores Pessoais. In: XX Simpósio Brasileiro de Informática na Educação - SBIE, 2009, Florianópolis-SC, 2009-b.

VICARI, R. M.; GLUZ, J. C.; PASSERINO, L. M.; SANTOS, E. R.; PRIMO, T.; ROSSI, L. H. L.; BORDIGNON, A.; BEHAR, P. A.; FERREIRA FILHO, R. C. M.; ROESLER, V. The OBAA Proposal for Learning Objects Supported by Agents. In: Joint International Workshop on Multi-Agent Systems for Education and Interactive Entertainment MASEIE/AAMAS, 2010, Toronto. Proceedings of Joint International Workshop on Multi-Agent Systems for Education and Interactive Entertainment, 2010. p. 43-46.

VIDEOLAN. VLC media player: The cross-platform open-source multimedia framework, player and server. 2010. Disponível em: <<http://www.videolan.org/vlc/>>. Acesso em: nov. 2010.

W3C. HTML 4.01 Specification. W3C Recommendation. 1999. Disponível em: <<http://www.w3.org/TR/html401/>>. Acesso em: jul. 2010.

W3C. XHTML 1.0 - The Extensible HyperText Markup Language (Second Edition). W3C Recommendation. 2002. Disponível em: <<http://www.w3.org/TR/xhtml1/>>. Acesso em: jul. 2010.

W3C. Extensible Markup Language (XML). 2003-a. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: jul. 2010.

W3C. Graphics on the Web. 2003-b. Disponível em: <<http://www.w3.org/Graphics/>>. Acesso em: jul. 2010.

W3C. JPEG. 2003-c. Disponível em: <<http://www.w3.org/Graphics/JPEG/>>. Acesso em: jul. 2010.

W3C. CSS Mobile Profile 2.0. W3C Candidate Recommendation. 2008-a. Disponível em: <<http://www.w3.org/TR/css-mobile/>>. Acesso em: ago. 2010.

W3C. Mobile Web Best Practices 1.0, Basic Guidelines. W3C Recommendation. 2008-b. Disponível em: <<http://www.w3.org/TR/mobile-bp/>>. Acesso em: jul. 2010.

W3C. About the World Wide Web Consortium. 2009-a. Disponível em: <<http://www.w3.org/Consortium/about-w3c.html>>. Acesso em: dez. 2009.

W3C. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. W3C Candidate Recommendation. 2009-b. Disponível em: <<http://www.w3.org/TR/CSS21/>>. Acesso em: nov. 2009.

W3C. HTML5. 2010-a. Disponível em: <<http://www.w3.org/TR/html5/>>. Acesso em: dez. 2010.

W3C. Scalable Vector Graphics (SVG). XML Graphics for the Web. 2010-b. Disponível em: <<http://www.w3.org/Graphics/SVG/>>. Acesso em: jul. 2010.

WAP FORUM. Wireless Application Environment Defined Media Type. 2001. Disponível em: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-237-waemt-20010515-a.pdf>>. Acesso em: jul. 2010.

WILKINSON, Jim; DEVLIN, Bruce. The Material Exchange Format (MXF) and its Application. In SMPTE Journal, Set. 2002. Pg. 378 a 384.