

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

DESENVOLVIMENTO DE SERVIDOR DE RSS PARA TV
DIGITAL INTERATIVA

EVERTON SCHROEDER

BLUMENAU
2007

2007/2-12

EVERTON SCHROEDER

DESENVOLVIMENTO DE SERVIDOR RSS PARA TV

DIGITAL INTERATIVA

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Mauro Marcelo Mattos, Orientador

BLUMENAU
2007

2007/2-12

**DESENVOLVIMENTO DE SERVIDOR RSS PARA TV
DIGITAL INTERATIVA**

Por

EVERTON SCHROEDER

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente:

Prof. Mauro Marcelo Mattos, Dr. – Orientador, FURB

Membro:

Prof. Francisco Adell Péricas, M.Sc. – FURB

Membro:

Prof. Dalton Solano dos Reis, M.Sc. – FURB

Blumenau, 07 de novembro de 2007.

AGRADECIMENTOS

Primeiramente a Deus, muitas vezes lembrado apenas em momentos de dificuldade, mas contando com Sua força e graça tudo é possível.

A meus pais, eternos modelos de caráter e integridade para o meu crescimento como ser humano e cidadão, a quem muito devo por ter chegado até aqui.

Aos meus amigos e demais colegas; tanto os que disseram que esta seria uma difícil tarefa para manter-me alerta, quanto os que apoiaram e passaram-me tranquilidade. Uma menção especial aos colegas de curso ou de profissão que de alguma forma tenham compartilhado experiências ou quaisquer outros recursos, sempre bem vindos nessas horas.

E finalmente, mas não menos importante, ao meu orientador e professor Mauro, por apresentar idéias, perspectivas e também por ditar o ritmo de trabalho, sua ajuda foi fundamental para a execução deste projeto.

Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá.

Ayrton Senna da Silva

RESUMO

O sistema de transmissão e recepção de sinal televisivo em formato digital propicia melhor qualidade de som e imagem, possibilitando ainda o tráfego de dados juntamente com o sinal. Considerando o fato de que alguns receptores de TV digital, quando associados com um meio de transmissão de dados como um modem podem interagir com o provedor de serviço de difusão através de um canal de retorno. Este trabalho procura demonstrar as características e funcionalidades deste recurso, através de um sistema de fornecimento de notícias ao telespectador por meio de mensagens enviadas via RSS.

Palavras-chave: Televisão digital. Interatividade. Canal de retorno.

ABSTRACT

The broadcasting and receiving television system using the digital format provides better audio and video quality, making possible data streaming combined with the TV signal. Considering the fact that some digital TV receivers will come with a modem integrated and it can interact with broadcast television provider through a return channel. This work aims to demonstrate the properties and functionalities that can be found in this resource, developing a news provider system by messages sent via RSS.

Key-words: Digital television. Interactivity. Return channel.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo de um sistema de TV digital interativa (TVDI).....	13
Figura 2 – Infra-estrutura de distribuição da TVDI.....	18
Figura 3 – Representação de conexão com o canal de retorno no <i>middleware</i> MHP	21
Figura 4 – Exemplo de um <i>set-top box</i>	23
Figura 5 – Módulos de um <i>set-top box</i>	24
Figura 6 – Camadas da arquitetura para um sistema de TVDI.....	26
Figura 7 – Modelo da arquitetura MHP.....	30
Figura 8 – Pilha <i>desoftware</i> do MHP	31
Figura 9 – Camadas da interface gráfica de uma aplicação MHP.....	32
Figura 10 – Ciclo de vida de uma <i>Xlet</i>	34
Quadro 2 – Elementos requeridos para a representação de um canal em um arquivo RSS	36
Quadro 3 – Elementos opcionais para representação de um canal em um arquivo RSS	36
Quadro 4 – Elementos opcionais para representação de um canal em um arquivo RSS	37
Quadro 5 – Exemplo de um <i>feed</i> RSS	37
Figura 11 – Ambiente de emulação OpenMHP	39
Figura 12 – Demonstração do aplicativo de comércio eletrônico	43
Figura 13 – Diagrama de casos de uso	46
Quadro 6 – Caso de uso: Efetuar <i>login</i> no sistema.....	47
Quadro 7 – Caso de uso: Configurar opções de funcionamento	47
Quadro 8 – Caso de uso: Requisitar o recebimento de notícias no formato RSS.....	47
Quadro 9 – Caso de uso: Consulta detalhes da manchete	48
Quadro 10 – Caso de uso: Adicionar notícia aos favoritos	48
Quadro 11 – Caso de uso: Excluir notícia dos favoritos	48
Figura 14 – Diagrama de classes	50
Figura 15 – Diagrama de estados	51
Figura 16 - Visão geral do modelo	53
Quadro 12 – Tratamento dos eventos do controle remoto.....	55
Quadro 13 – Tratamento dos eventos do controle remoto.....	56
Quadro 14 – Implementação de entrada de texto no teclado numérico.....	57
Quadro 15 – Conexão com canal de retorno	58
Quadro 16 – Interpretação do arquivo XML.....	59

Figura 17 – Tela inicial da aplicação.....	60
Figura 18 – Tela de <i>login</i>	61
Figura 19 – <i>Login</i> inválido	62
Figura 20 – Painel de configurações.....	63
Figura 21 – Tela de exibição das manchetes	64
Figura 22 – Painel de exibição do resumo da notícia.....	64
Figura 23 – Tela com os itens gravados nos favoritos	65

LISTA DE SIGLAS

8-VSB – *8 Vestigial Side Band*

ADSL – *Assimetric Digital Subscriber Line*

API – *Application Programming Interface*

ARIB – *Association of Radio Industries and Businesses*

ATSC – *Advanced Television Systems Comitee*

AWT – *Abstract Window Toolkit*

BCC – Curso de Ciências da Computação – Bacharelado

BML – *Broadcast Markup Language*

COFDM – *Coded Orthogonal Frequency Division Multiplexing*

COM-TV – Comissão Assessora para Assuntos de Televisão

CPQD – Centro de Pesquisa e Desenvolvimento em Telecomunicações

DASE – *Digital TV Application Software Environment*

DAVIC – *Digital Audio Visual Council*

DiBEG – *Digital Broadcasting Experts Group*

DOM – *Document Object Model*

DNS – *Domain Name System*

DSC – Departamento de Sistemas e Computação

DVB – *Digital Video Broadcasting*

EPG – *Electronic Programming Guide*

FINEP – Financiadora de Estudos e Projetos

HAVi – *Home Audio/Video Interoperability*

HDTV – *High Definition Television*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IEEE – *Institute of Electrical and Eletronics Engineers*

ISDB – *Integrated Services Digital Broadcasting*

JMF – *Java Media Framework*

JVM – *Java Virtual Machine*

MHP – *Multimedia Home Platform*

PCMCIA – *Personal Comupter Memory Card International Association*

PID – *Packet Identification Number*

POO – *Programação Orientada a Objetos*

QAM – *Quadrature Amplitude Modulation*

QPSK – *Quadrature Phase Shift Keying*

RSS – *Rich Site Summary*

SAX - *Simple API for XML*

SMTP – *Simple Mail Transfer Protocol*

TCP/IP – *Transmission Control Protocol/Internet Protocol*

TVDI – *Televisão digital interativa*

UML – *Unified Modeling Language*

VOD – *Video on Demand*

XML – *eXtensible Markup Language*

SUMÁRIO

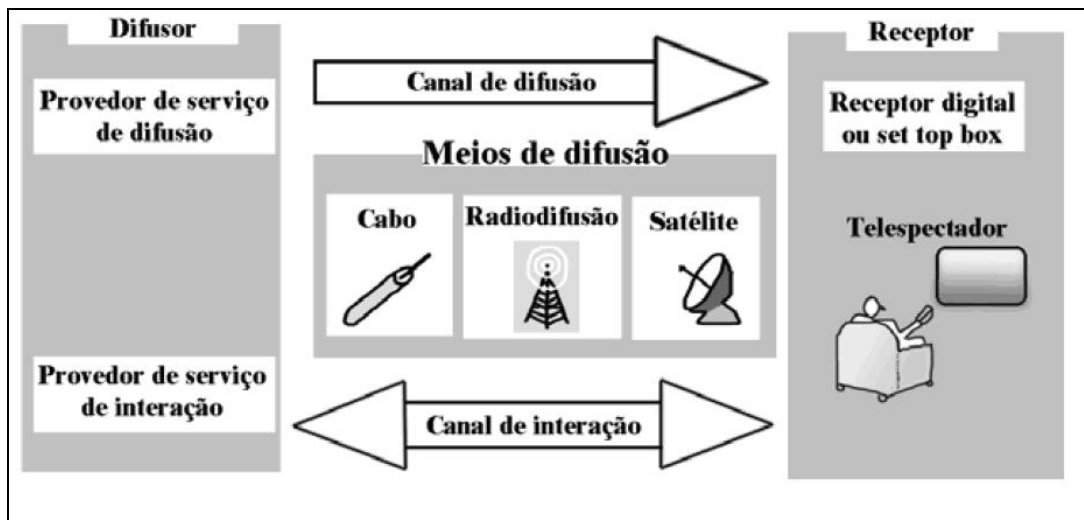
1	INTRODUÇÃO	13
1.1	OBJETIVOS DO TRABALHO	14
1.2	ESTRUTURA DO TRABALHO.....	15
2	FUNDAMENTAÇÃO TEÓRICA.....	16
2.1	TV DIGITAL	16
2.1.1	Aplicações e serviços	17
2.1.2	Infra-estrutura.....	18
2.1.2.1	Transmissão	19
2.1.2.2	Modulação	19
2.1.2.3	Digital Storage Media - Command and Control (DSM-CC).....	20
2.1.2.3.1	Canal de retorno	20
2.1.2.3.2	Sistema de acesso condicional	22
2.2	SET-TOP BOX	22
2.2.1.1	Módulos e funcionalidades	23
2.2.1.2	Interfaces de comunicação.....	25
2.3	PADRÕES DE TV DIGITAL E MIDDLEWARES.....	25
2.3.1	ATSC-DASE.....	26
2.3.2	DVB-MHP	27
2.3.3	DIBEG-ISDB	27
2.3.4	O <i>middleware</i> MHP	28
2.3.4.1	Perfis de execução	28
2.3.4.2	Arquitetura	29
2.3.4.3	Pilha de software.....	30
2.3.4.4	Interface gráfica para o usuário	31
2.4	BIBLIOTECAS DE SUPORTE A MIDDLEWARES JAVATV E XLETS	33
2.5	RSS.....	34
2.5.1	Vantagens de utilização do RSS	35
2.5.2	Estrutura de um arquivo RSS.....	35
2.6	EMULADORES	38
2.6.1	Ambiente de emulação OpenMHP.....	38
2.7	O SISTEMA BRASILEIRO DE TV DIGITAL	39

2.7.1 Histórico.....	40
2.7.2 Objetivos do programa.....	41
2.8 TRABALHOS CORRELATOS.....	42
2.8.1 Aplicação de comércio via TV digital	42
2.8.2 Interpretador de documentos RSS para sistemas de televisão	43
2.8.3 Portal para aplicações colaborativas em TV digital interativa.....	44
3 DESENVOLVIMENTO DA APLICAÇÃO	45
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO	45
3.2 ESPECIFICAÇÃO	45
3.2.1 Casos de uso.....	46
3.2.2 Diagrama de classes	49
3.2.3 Diagrama de estados	51
3.3 IMPLEMENTAÇÃO	52
3.3.1 Técnicas e ferramentas utilizadas.....	52
3.3.1.1 Contexto da aplicação.....	52
3.3.1.2 Ambiente de desenvolvimento: Eclipse.....	53
3.3.1.3 Interface gráfica e tratamento de eventos	54
3.3.1.4 O canal de retorno.....	58
3.3.1.5 Interpretação do arquivo XML que contém o feed.....	59
3.3.2 Operacionalidade da aplicação.....	60
3.4 RESULTADOS E DISCUSSÃO	65
4 CONCLUSÕES	67
4.1 EXTENSÕES	68
REFERÊNCIAS BIBLIOGRÁFICAS	69

1 INTRODUÇÃO

A transmissão de imagens pela televisão encontra-se atualmente em uma fase de transição, com a introdução do sistema de televisão digital no Brasil, que estará abrindo um vasto leque de possibilidades e novos serviços à disposição das pessoas. De acordo com Monteiro (2004, p. 7), “... o principal diferencial da televisão digital em relação à sua precursora analógica é a possibilidade do transporte de dados e de aplicações além de áudio/vídeo.”.

Como pode ser visto na Figura 1, o sinal digital de televisão a ser difundido pode ter a recepção via satélite, cabo ou radiodifusão, enviando imagem, som e até dados por esse mesmo sinal. Ao ser recebido por uma antena ou qualquer outro meio, esse sinal é decodificado por um receptor, que pode estar embutido na própria televisão, ou, nos casos de ainda estar utilizando-se um aparelho analógico, faz-se necessária a instalação de um terminal de acesso, também conhecido como *set-top box*. Esses terminais podem processar conteúdo televisivo, executar algumas aplicações e manipular gráficos transmitidos juntamente com o sinal, permitindo certa interatividade local (MONTEZ; BECKER, 2005, p. 106).



Fonte: Montez e Becker (2005, p. 106).

Figura 1 – Modelo de um sistema de TV digital interativa (TVDI)

Segundo Piccioni e Montez (2004, p. 2), o conceito de interatividade pode tornar-se muito mais amplo quando o receptor possuir um recurso chamado canal de retorno. Com ele, ao invés de apenas receber sinais da emissora, o telespectador pode enviar dados de volta para a mesma, tornando possível a criação de uma imensa gama de novos serviços, como acessar a Internet, *home-banking*, jogos, bate-papo e vídeo sob demanda - *Video on Demand* (VOD).

Bezerra (2004, p. 6) afirma que “... toda essa interatividade digital propiciada por esse

terminal de serviços pretende explorar a ligação entre o envolvimento oferecido pela televisão analógica atualmente existente com o acesso diversificado de informação característico da Internet.”. Essas qualidades citadas têm como objetivo propiciar uma programação com conteúdo de melhor qualidade, interativo e que forneça exploração e imersão pelo usuário a um novo meio de comunicação.

Montez e Becker (2005, p. 52) definem que as aplicações para TV digital podem ser divididas em duas categorias: reativas e interativas. As que se enquadram no padrão reativo são aquelas que não possuem canal de retorno, apenas recebem o sinal de televisão digital e são incapazes de enviar qualquer resposta para a fonte transmissora do sinal, impossibilitando assim a interatividade. Já as interativas são as que possuem canal de retorno, podendo transmitir informações para a fonte transmissora do conteúdo digital, permitindo assim interação real com a programação que está sendo exibida.

Com o aumento do volume de informações vindas de vários lugares do planeta, existe a necessidade destas serem veiculadas em vários tipos de mídia. Com a popularização da Internet, novos portais de informação estão surgindo a cada dia. Para reduzir o tempo perdido com a visitação de páginas na Internet em busca de conteúdo, foi desenvolvido um serviço chamado *Rich Site Summary* (RSS). Este serviço está baseado no uso de servidores que catalogam o conteúdo de diversos outros servidores e disponibilizam para o usuário um *link* para o *site* contendo a informação atualizada ou um recorte da manchete publicada. Com isso, o usuário pode receber apenas informações que forem de seu interesse (ALECRIM, 2005).

Como o serviço de TV digital encontra-se ainda em fase de implantação no país, as primeiras implementações focadas nesta área são executadas em ambientes conhecidos como emuladores. De acordo com Morris (2002), os ambientes de emulação mais conhecidos são o XletView e o OpenMHP.

Assim, pretende-se desenvolver neste trabalho um protótipo para TVDI, executando em um ambiente de emulação, onde o telespectador conecta-se a um servidor por meio do canal de retorno requisitando o recebimento de notícias enviadas por RSS.

1.1 OBJETIVOS DO TRABALHO

Este trabalho tem como objetivo principal demonstrar o potencial do uso da TVDI para o desenvolvimento de aplicações interativas utilizando o canal de retorno. Para atingir este

objetivo, é desenvolvido um protótipo de aplicação para o ambiente de TVDI utilizando o padrão de *middleware Multimedia Home Platform* (MHP) que disponibiliza ao telespectador a possibilidade de requisitar a recepção de notícias enviadas por RSS.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em quatro capítulos. No capítulo 2 é descrita a fundamentação teórica utilizada como referência para este trabalho, apresentando alguns conceitos sobre TVDI e seus principais padrões, bibliotecas de suporte ao desenvolvimento de aplicações interativas para este ambiente e o *middleware* MHP, o protocolo RSS, o Sistema Brasileiro de TV Digital, emuladores para simulação do ambiente de TVDI. Ao final desta seção são apresentados alguns trabalhos correlatos.

O capítulo 3 descreve o desenvolvimento do trabalho e a especificação da aplicação, representada por diagramas de casos de uso, estados e também o diagrama de classes. A seguir, encontram-se descritas algumas rotinas implementadas e também um guia operacional da aplicação.

Por fim, o capítulo 4 apresenta as conclusões obtidas com o trabalho juntamente com algumas sugestões para a realização de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos relacionados ao trabalho, os quais são: TV digital, *set-top box*, canal de retorno, *middlewares* e os principais padrões existentes, a biblioteca JavaTV de suporte a desenvolvimento de *middlewares*, o serviço RSS, emuladores, com destaque para o ambiente OpenMHP. Ao final deste capítulo, é apresentado o Sistema Brasileiro de TV Digital e três trabalhos correlatos com o tema em questão.

2.1 TV DIGITAL

Montez e Becker (2005, p. 28) definem a televisão digital como “... um sistema de transmissão, recepção e processamento de sinais em formato digital, que permitem uma alta definição de som e imagens além de grande interatividade.”. A digitalização na transmissão dos sinais de TV pode oferecer diversos benefícios.

O processo de codificação e compressão de sinais de áudio e vídeo em formatos especiais, como por exemplo, MPEG-2, desenvolvido pela *Moving Pictures Expert Group*, para codificar e transmitir sinais de áudio e vídeo de forma digital vem sendo amplamente utilizado, possibilitando cinco vezes ou mais canais numa mesma largura de banda. Assim, as emissoras de sinais de TV digital aumentam seu potencial de receita com a distribuição de mais conteúdo e de eventos pagos – conhecidos como *pay-per-view*. O usuário final desfruta de uma qualidade de áudio e vídeo superior em comparação com as televisões analógicas, incluindo áudio digital *surround*¹ e vídeo em *widescreen*².

Segundo Unicamp (2006), os preços de televisores digitais são proibitivos para a maioria dos telespectadores. A fim de atender esses telespectadores, sem a necessidade de descartar os aparelhos de televisão analógicos, serão comercializados receptores denominados *set-top boxes* que serão responsáveis pelo processamento do sinal digital, interação com o telespectador e pela entrega de conteúdo audiovisual para o mesmo através de uma televisão

¹ *Surround* é um sistema de áudio que possui o som dividido em quatro canais e mais um para sons graves. (SILVA, 2006).

² Videolar (2002) define *widescreen* como o aspecto da tela de um televisor com a razão de 16:9, ou seja, dezesseis unidades de largura por nove unidades de altura.

analógica comum.

Um grande benefício da TVDI é a possibilidade de oferecimento, graças ao canal de dados, de novas aplicações e serviços, que são descritos na subseção a seguir.

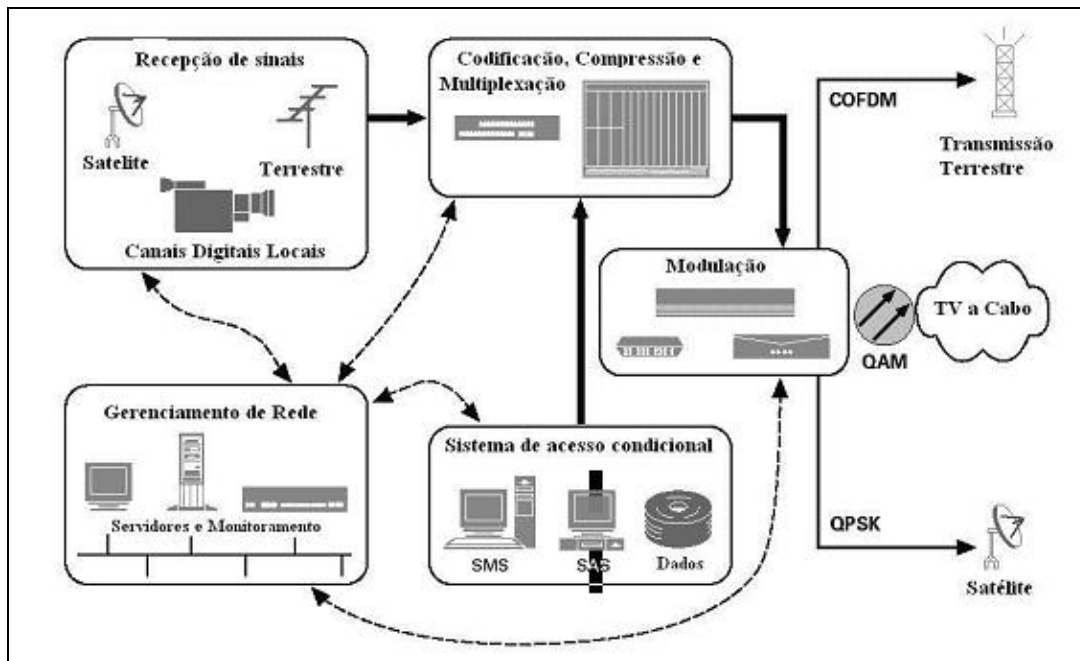
2.1.1 Aplicações e serviços

Andreatta (2006, p. 4) diz que diversas aplicações e serviços são possíveis na TVDI e cita alguns dos serviços mais utilizados atualmente:

- a) *Electronic Program Guide* – EPG: é um guia de programação eletrônica que fornece informações detalhadas sobre o conteúdo dos programas em diversos canais. Com a disponibilidade de centenas de canais e programas sendo transmitidos, o único método de busca viável é através de uma *engine* de procura na qual podem ser informadas palavras chaves que retornam uma lista de programas que combinam com o critério de busca;
- b) *T-Commerce*: modalidade de comércio eletrônico, já que o apelo comercial fortemente presente na TVDI permite que aplicações comerciais sejam transmitidas com o programa visto pelo espectador permitindo ao mesmo a possibilidade de efetuar compras de produtos relacionados ou não ao programa visto;
- c) *Video on Demand* – VOD: serviço que permite ao telespectador escolher qual e quando deseja assistir determinado programa, fazendo com que a TV se transforme em uma locadora doméstica;
- d) notícias interativas: durante a transmissão do programa de notícias, manchetes em texto podem correr horizontalmente na parte superior ou inferior da tela, possibilitando ao telespectador selecionar a de sua preferência e buscar maiores informações para serem mostradas na tela, enquanto que o programa de notícias continua sendo visto em uma janela que não ocupa a tela inteira;
- e) programação educacional interativa: através da TVDI vários programas podem ser desenvolvidos para auxiliar o ensino de crianças e adultos. Aulas podem ser ministradas à distância, com o aluno em sua casa ou na escola, sendo as aulas ao vivo ou com conteúdo gravado, com horário pré-determinado para transmissão ou com o uso de VOD.

2.1.2 Infra-estrutura

De forma simplificada, distinguem-se três papéis num ambiente de TVDI: transmissores, telespectadores e provedores de conteúdo. A Figura 2 mostra como essas três partes relacionam-se entre si.



Fonte: Andreatta (2006, p. 5).

Figura 2 – Infra-estrutura de distribuição da TVDI

O provedor de conteúdo gera o som, imagem e aplicações que compõem os seus programas e entrega para um ou mais transmissores. O transmissor faz a multiplexação do que é recebido dos provedores de conteúdo num fluxo de transporte e envia o sinal para os telespectadores. A diferença desse transmissor da TVDI para o da TV analógica é que ele tem a possibilidade de transmitir mais de um serviço no espectro de frequência onde antigamente passava apenas um serviço. Ele pode ser caracterizado pelo padrão de TVDI que utiliza, pelos grupos de telespectadores que consegue atingir e o tipo de transmissão (terrestre, cabo ou satélite). O telespectador continua a assistir à TV, mas utilizando de um aparelho denominado *set-top box*, que além de decodificar o sinal televisivo tem a possibilidade de interagir com a programação (TAKEGUMA, 2005, p. 9).

2.1.2.1 Transmissão

As informações geradas pela plataforma incluem dados sobre os canais e programas transmitidos, dados gerados para controle de acesso aos canais e eventos produzidos a partir de interações de usuários via um canal de retorno (processados por uma plataforma específica). As fontes de áudio e vídeo codificadas no formato MPEG-2, são multiplexadas em conjunto com informações de controle em um único fluxo contínuo de dados. As informações de controle de acesso são usadas nesse processo para encriptação do conteúdo específico de canais de acesso restrito. O fluxo de dados gerado pelo multiplexador MPEG-2 ainda passa por outras etapas para que esteja preparado para ser transmitido. Um processo de prevenção e correção de erros é primeiramente aplicado. Posteriormente, ocorre a modulação do sinal (conversão para sinais analógicos para que possam ser transmitidos por ondas de rádio, por exemplo) e por fim, a conversão destes sinais para a frequência em que serão transmitidos (ANDRADE, 2004, p. 10).

2.1.2.2 Modulação

Haykin apud Montez e Becker (2005, p. 71) afirmam que na TV digital, áudio, vídeo e dados precisam ser transportados desde sua origem até a casa do usuário (podendo passar ou não por estações intermediárias). Nesse sentido, um sistema de comunicação é usado para esse transporte. Contudo, as informações não podem ser enviadas diretamente pelo sistema de comunicação sem antes sofrer uma modulação no envio, e uma demodulação na recepção.

A modulação oferece três benefícios:

- a) desloca o conteúdo espectral de um sinal de mensagens (dados e mídias digitais, no caso da TV digital) para dentro da faixa de frequência operacional de um enlace de comunicação;
- b) permite colocar o conteúdo do sinal de mensagens numa forma menos vulnerável a ruído ou interferência;
- c) possibilita multiplexar o enlace de comunicação, ou seja, compartilhá-lo, transmitindo simultaneamente dados oriundos de fontes independentes.

O sinal a ser transmitido, que na saída do multiplexador é formado por sinais binários 0 e 1, é modulado, isto é, o sinal é acoplado a uma portadora, onde são acrescentados estados

para facilitar detecção de erros e aumento na taxa de transmissão. Os métodos de modulação digital com maior utilização atualmente são: o *Quadrature Amplitude Modulation* (QAM), *Quadrature Phase Shift Keying* (QPSK), *Coded Orthogonal Frequency Division Multiplexing* (COFDM) e *8-Vestigial Side Band* (8-VSB) (ANDREATA, 2006, p. 6).

2.1.2.3 *Digital Storage Media - Command and Control* (DSM CC)

A especificação DSM-CC define uma série de protocolos que fornece funções de controle sobre fluxos de transmissão MPEG em ambientes de rede heterogêneos. Foi inicialmente concebida como parte da especificação do padrão MPEG-2, consistindo em um protocolo simples cujo objetivo era possibilitar o controle remoto sobre um fluxo MPEG (uma funcionalidade similar a de um aparelho de vídeo-cassete executando suas operações sobre uma fita VHS na rede). Desde então a especificação evoluiu bastante, gerando aplicações na distribuição de serviços audiovisuais interativos, como vídeo sob demanda e TV digital. (ANDRADE, 2004, p. 19).

Andreatata (2006, p. 10) afirma que o DSM-CC é independente da camada de transporte. O ponto chave é que cada protocolo DSM-CC pode ser usado de forma independente ou em cooperação com outros. Estes protocolos cobrem as seguintes áreas:

- a) gerenciamento de seções e recursos da rede;
- b) configuração de clientes;
- c) transferência de dados aos clientes;
- d) gerenciamento de fluxos de vídeo;
- e) aplicações para o controle dos serviços disponibilizados;
- f) carrossel de dados e objetos para aplicações transmitidas por difusão.

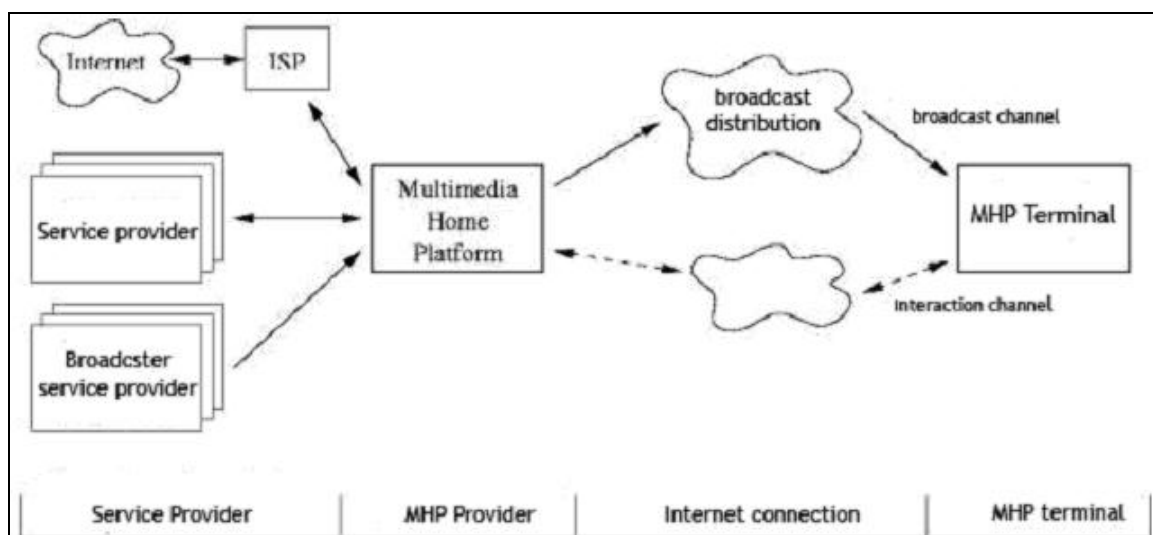
2.1.2.3.1 Canal de retorno

Para os telespectadores, apenas a adição de novos conteúdos que podem ser vistos juntamente com a imagem televisiva não é o suficiente para garantir sucesso com a experiência em termos de interatividade. Takeguma (2005, p. 10) salienta que a existência de um canal de retorno, que é a comunicação do *set-top box* com o mundo externo, é responsável por definir o grau de interatividade, que pode ser classificar em três níveis:

- a) interatividade local, em que o telespectador tem a possibilidade apenas de navegar pelos dados que são transmitidos no fluxo de transporte;
- b) interatividade em uma direção, na qual as ações do usuário são armazenadas em *buffer* e enviadas a um servidor de tempos em tempos;
- c) interatividade em duas direções, na qual o telespectador tem acesso direto à Internet, que é o maior nível de interação e possibilita aplicações como jogos em tempo real, navegação na Internet, vídeo sob demanda, etc.

Segundo Andreatta (2006, p. 13), a conexão com um canal de retorno pode ser uma conexão permanente (*Ethernet, Assymmetric Digital Subscriber Line (ADSL) ou cable modem*) ou temporária, como as conexões discadas. Especificações de sistemas de TVDI costumam exigir o suporte ao protocolo *Transmission Control Protocol/Internet Protocol (TCP/IP)* no canal de retorno. Para conexões permanentes é utilizado parte do já conhecido pacote `java.net`, importado do Java tradicional, onde encontram-se classes como a `java.net.InetAddress` (para a resolução de nomes através de *Domain Name System – DNS*), `java.net.Socket` (para gerenciar conexões) e `java.net.URLConnection` para o acesso de páginas no formato *HyperText Transfer Protocol (HTTP)*. Atualmente somente é obrigatório o suporte ao protocolo DNS, sendo que protocolos como o *Simple Mail Transfer Protocol (SMTP)* devem ser implementados pela aplicação.

Na Figura 3 é representado o esquema de funcionamento do sistema de TVDI com o uso do canal de difusão unidirecional e do canal de retorno bidirecional.



Fonte: Casas (2006, p. 20).

Figura 3 – Representação de conexão com o canal de retorno no *middleware* MHP

Conforme Casas (2006, p.21), as decisões em torno da escolha de uma tecnologia para o canal de retorno não são apenas de ordem técnica, mas também pela influência no custo

final de fabricação do *set-top box* dependendo da tecnologia empregada.

Takeguma (2005, p. 10) diz que a transmissão de TVDI por cabo tipicamente tem interatividade em duas direções, já que sua estrutura física permite que o caminho de retorno percorra o mesmo cabo que traz os dados. Já as transmissões por satélite e terrestre dependem de uma conexão à parte (via *modem*, por exemplo).

2.1.2.3.2 Sistema de acesso condicional

Acesso condicional é o processo que determina quais terminais de TVDI podem fornecer certas programações distribuídas pela emissora para os telespectadores que tenham previamente comprado estes serviços. Este sistema já era uma parte essencial nos negócios de TV analógica por assinatura e, portanto, já está sendo de vital importância para o desenvolvimento do sucesso nos negócios em TVDI (VIEIRA, 2005, p. 41).

Andreatta (2006, p. 13) diz que o controle de acesso à programação requer o uso de duas principais tecnologias. Primeiramente, os meios de encriptação e decriptação da transmissão e as mensagens de controle para que esta decriptação seja realizada somente nos *set-top boxes* de clientes que pagaram pelo serviço, o chamado Sistema de Acesso Condicional. Este sistema acrescenta palavras de controle no encriptador, e garante que somente os decriptadores com permissão recebam esta palavra de controle para desembaralhar o fluxo de dados MPEG.

2.2 SET-TOP BOX

Montez e Becker (2005) definem que os *set-top boxes* (Figura 4) são equipamentos desenvolvidos para a decodificação do sinal digital, responsáveis por processar som, imagem e dados para que possam ser exibidos em aparelhos de televisão convencionais. Para permitir ao telespectador a interação com os serviços disponíveis, esses equipamentos devem possuir capacidade de processamento de conteúdo, comuns aos computadores, como processador, memória, disco rígido e *modem*, semelhantes a computadores. De acordo com Andreatta (2006, p. 17), todo este pacote possui exigências em relação à robustez e confiabilidade, pois os telespectadores não estão acostumados com mensagens de erro durante a programação

televisiva ou com a necessidade de reiniciar o aparelho durante a exibição de um filme. Devem ainda ser capazes de lidar com várias tarefas simultaneamente, como enviar e receber dados, salvar conteúdo localmente, etc.



Fonte: Guimarães (2005, p. 18).

Figura 4 – Exemplo de um *set-top box*

2.2.1.1 Módulos e funcionalidades

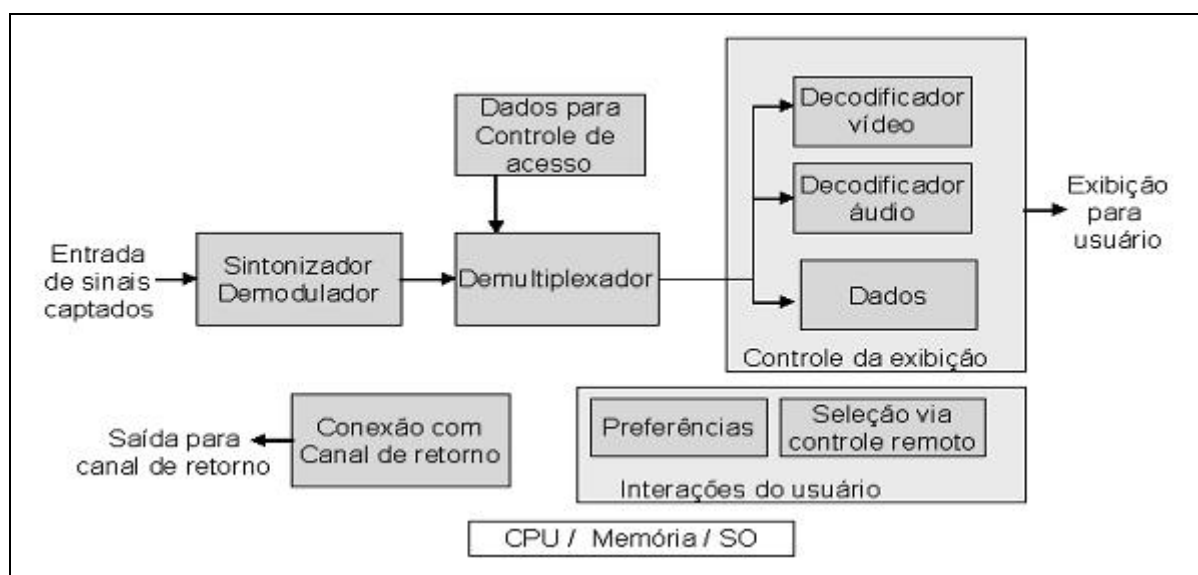
As principais funcionalidades de um *set-top box* consistem em processar o sinal televisivo que é recebido, verificação de direitos de acesso a conteúdo, saída de áudio e vídeo, execução de aplicações e interação. Zhang (2003, p. 5) descreve cada uma dessas funcionalidades:

- a) processamento do sinal recebido: a capacidade de processar o sinal transmitido pela difusora de conteúdo televisivo é um requisito fundamental. O sinal televisivo ao ser transmitido no ar está em forma analógica com a banda normal de canal de 8MHz. O *set-top box* então precisa selecionar uma determinada frequência de acordo com um canal, função esta denominada sintonização. A saída desta sintonização precisa então ser demodulada para extrair áudio, vídeo e as demais informações, além de corrigir erros de integridade nos dados recebidos;
- b) verificação de direitos de acesso: o controle de acesso a informações é a base para quase todos os modelos de negócios na TVDI, desde serviços relacionados à Internet até vídeo sob demanda. O *set-top box* desembaralha o fluxo de transporte baseado em um *Packet Identification Number* (PID), que define a qual programa ou serviço este *packet* pertence. Somado a isso, os dados contendo mensagens de acesso condicional embutidas no fluxo transportado com o sinal televisivo também

são extraídos durante a demultiplexação e repassados ao subsistema de acesso condicional para verificação. Certos modelos de segurança são aplicados para proteger o conteúdo de visualizações não-autorizadas;

- c) saída de áudio e vídeo: o fluxo de informações transportadas em MPEG-2 é separado e demultiplexado. Sendo assim, o sinal de vídeo precisa ser convertido para um formato suportado pelo monitor ou qualquer outro dispositivo que faça a exibição da imagem. O sinal de áudio não necessita de conversão e é encaminhado diretamente para o dispositivo de saída de áudio, tão logo seja feita a conversão de digital para analógico;
- d) processamento de aplicações e renderização: executa tanto aplicações residentes no *set-top box* quanto aplicações recebidas com o sinal televisivo. Após a multiplexação, os dados e aplicações são extraídos do fluxo de transporte e o gerenciador de aplicações é ativado para executar a aplicação recebida. O *set-top box* renderiza o resultado da execução no monitor da televisão juntamente com o vídeo;
- e) interação: a adição de um canal de interação dedicado faz com que o fato de assistir televisão seja muito mais do que assistir passivamente o conteúdo. Com a execução das aplicações, os telespectadores podem navegar e selecionar o conteúdo que for de seu interesse, por meio de um controle remoto ou qualquer outra interface com o *set-top box*.

A Figura 5 ilustra os módulos de um *set-top box* e como eles relacionam-se entre si.



Fonte: Andreatta (2006, p. 15).

Figura 5 – Módulos de um *set-top box*

2.2.1.2 Interfaces de comunicação

Os *set-top boxes* podem fazer uso de várias outras interfaces para a comunicação com o mundo externo. Em SUN MICROSYSTEMS (2001) são citados alguns exemplos de interfaces, entre as quais estão:

- a) IEEE-1284 (Interface Paralela): utilizada para conexão a impressoras;
- b) RS-232 (Interface Serial): utilizada para conectar computadores e *modems*;
- c) USB (*Universal Serial Bus*): utilizada para conectar teclados, *joysticks*, *mouses* e vários outros equipamentos ao *set-top box*;
- d) IEEE-1394 (*Firewire*): conexão de alta velocidade para filmadoras e outros dispositivos áudio-visuais;
- e) 10 Base-T (*Ethernet*): conexão em rede local com computadores e impressoras;
- f) PCMCIA tipo II – Vários usos como, por exemplo, estender a capacidade de memória do *set-top box*, deciframento de acesso condicional para serviços de várias redes diferentes, aumentar a capacidade de armazenamento e adicionar novos sintonizadores ao *set-top box*;
- g) *SmartCards*: permite acesso a diversos serviços de várias redes, além do uso em comércio eletrônico;
- h) Controles remotos e teclados sem fio.

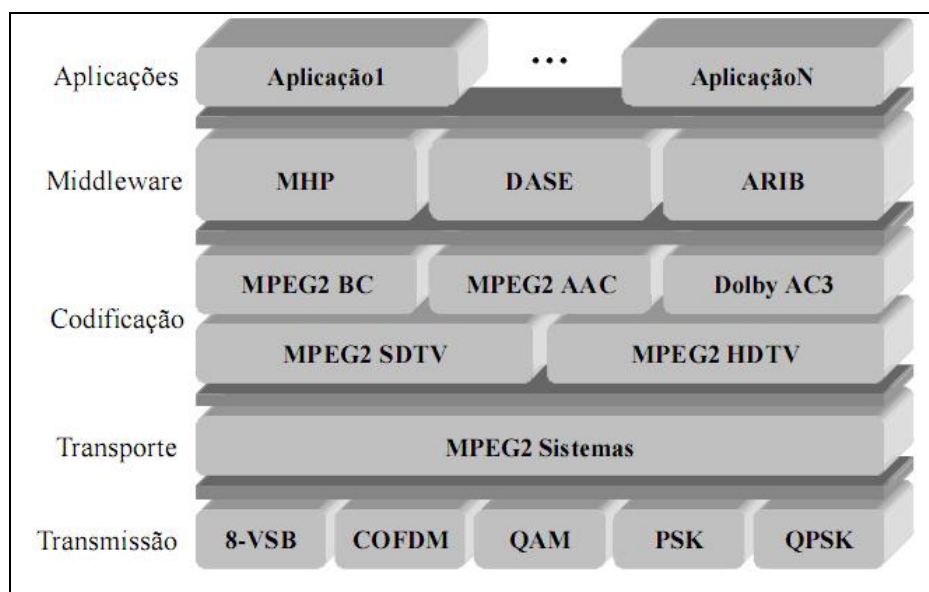
2.3 PADRÕES DE TV DIGITAL E MIDDLEWARES

Montez e Becker (2005, p.33) afirmam que “há algum tempo a comunidade que desenvolve tecnologia para TV digital percebeu que provedores de serviços não iam ter (sic) sucesso comercial se tivessem que desenvolver serviços interativos que não fossem portáteis em *set-top boxes* de diferentes fabricantes”. Para cumprir este objetivo, a idéia de oferecer serviços padronizados para TVDI faz uso de uma camada de *middleware*, que intermedeia toda a comunicação entre a aplicação e o resto dos serviços oferecidos.

A camada de *middleware* (ver hierarquia de camadas na Figura 6) abstrai a interação entre aplicação e o sistema operacional. Com a ajuda do *middleware*, as aplicações no *set-top box* não precisam ter preocupação sobre os detalhes do sistema operacional nem sobre

configurações físicas. A camada de abstração reduz significativamente a complexidade do desenvolvimento de aplicações. A máquina virtual é uma forma importante de *middleware* no ambiente de um *set-top box*. Uma máquina virtual não abstrai apenas o comportamento de um *hardware*, como também define uma interface comum para as aplicações. A máquina virtual mais utilizada é a *Java Virtual Machine (JVM)*. (ZHANG, 2003, p.11).

De acordo com Becker et al. (2004), o uso do *middleware* facilita a portabilidade das aplicações, permitindo que sejam transportadas para qualquer receptor digital que suporte o *middleware* adotado. Essa portabilidade é primordial em sistemas de TV digital, assumindo o fato de que nem todos os receptores digitais sejam exatamente iguais.



Fonte: Fernandes et al. (2004, p.23).

Figura 6 – Camadas da arquitetura para um sistema de TVDI

Segundo Morris (2002), os três principais padrões para TV digital, juntamente com os *middlewares* desenvolvidos para cada padrão – ver Figura 6 – são: o americano denominado *Digital TV Application Software Environment (DASE)* da *Advanced Television Systems Comitee (ATSC)*, o padrão europeu denominado *Digital Video Broadcasting – Multimedia Home Platform (DVB-MHP)* e o padrão japonês denominado *Integrated Services Digital Broadcasting (ISDB)*. Entre eles, existem algumas diferenças quanto à definição do desenvolvimento de serviços.

2.3.1 ATSC-DASE

O padrão ATSC é desenvolvido por uma organização dos Estados Unidos que

desenvolveu padrões técnicos de transmissão terrestre (ATSC-T) e via cabo (ATSC-C). Tendo sido também definido um *middleware* para a plataforma de receptores, o DASE. O interesse principal em seu desenvolvimento era a transmissão de serviços de TV em alta definição, conhecido como *High Definition Television* (HDTV). Todavia existem aspectos, nos quais este padrão é pobre: no suporte de serviços interativos e na transmissão para dispositivos móveis. Esta última característica é facilmente entendida ao se analisar a malha de cabeamento nos Estados Unidos, mostrando o porquê de um maior investimento neste tipo de transmissão (COELHO, 2005, p. 4).

2.3.2 DVB-MHP

Montez e Becker (2005, p. 33) descrevem que o DVB é conhecido como o padrão europeu de televisão digital e tem como objetivo oferecer um ambiente de TV interativa, independente de hardware e software específicos, aberto e interoperável para receptores de TV digital. O padrão DVB é adotado em países da Europa, além da Austrália, Malásia, Hong Kong, Índia, África do Sul e diversos outros países fora da Europa. Para esses países, a televisão digital terrestre é um serviço pago, e os *set-top boxes* são subsidiados pelas operadoras de TV.

O *middleware* desenvolvido para esse padrão é chamado de MHP e busca oferecer um ambiente de TVDI independente de *hardware* e *software* específicos, aberto e interoperável para os *set-top boxes*. Seu ambiente de execução baseia-se no uso de uma máquina virtual Java e um conjunto de interfaces para programação de aplicações (*Application Programming Interface* – API) que possibilitam aos programas escritos em Java o acesso a recursos e facilidades do receptor digital de forma padronizada (BECKER et al., 2004).

2.3.3 DIBEG-ISDB

O padrão ISDB (ISDB apud DARÓS, 2004, p. 23) foi especificado em 1999 no Japão pelo grupo *Digital Broadcasting Experts Group* (DiBEG), criado em 1997, composto por várias empresas e operadoras de televisão. O objetivo do grupo DiBEG é promover e especificar o sistema de difusão terrestre de TV digital japonês. As maiores vantagens desse padrão são a grande flexibilidade de operação e potencial para transmissões móveis e também

para dispositivos portáteis.

Conforme Montez e Becker (2005, p. 34), o *middleware* deste padrão é o *Association of Radio Industries and Businesses* (ARIB). Esse *middleware* é formado por alguns padrões como o *Data Coding and Transmission Specification for Digital Broadcasting* (ARIB STD-B24) que define linguagem declarativa denominada *Broadcast Markup Language* (BML). Essa linguagem, baseada em *eXtensible Markup Language* (XML) é usada para especificação de serviços multimídia para TV digital. Outra especificação do *middleware* é o *Application Execution Engine Platform for Digital Broadcasting* (ARIB STD-B23). Essa especificação é baseada no DVB-MHP, e indica uma tendência do ARIB de tentar estabelecer uma conformidade com outros padrões de *middleware*.

2.3.4 O *middleware* MHP

O consórcio DVB definiu um padrão aberto para TV digital chamado MHP (MHP MULTIMEDIA HOME PLATFORM, 2003), que apresenta um conjunto de tecnologias para implementar serviços digitais multimídia, definindo protocolos de transporte, ciclos de vida das aplicações, modelos de sinalização, e segurança, *plug-ins*, entre outros elementos.

Conforme já foi citado, o ambiente de execução do MHP é baseado no uso de uma máquina virtual Java e um conjunto de interfaces de programação de aplicações (APIs). Montez e Becker (2005, p. 132) afirma que uma aplicação DVB usando API Java é denominada aplicação DVB-J, mas além do uso da API Java, o MHP 1.1 introduziu a possibilidade de usar uma linguagem de programação semelhante ao *HyperText Markup Language* (HTML), denominada DVB-HTML.

2.3.4.1 Perfis de execução

Bezerra (2004, p. 10) faz uma descrição da definição das áreas de atuação das aplicações e a capacidade do *set-top box* em operar com os diferentes tipos de funcionalidades habilitadas pelo padrão do consórcio DVB, que são chamados *Profiles*. Esta categorização está dividida em três perfis:

- a) *Enhanced Profile*: definido desde a primeira versão da especificação MHP, é o nível mais básico, oferecendo suporte para executar aplicações locais do *set-top*

box, ou recebidas via *broadcast*;

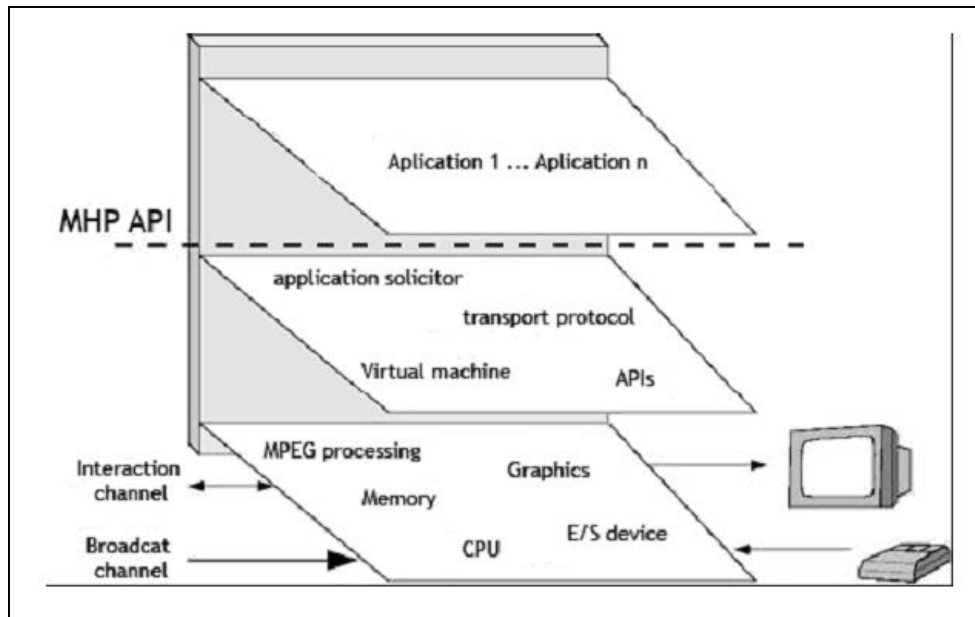
- b) *Interactive Profile*: também definido na versão inicial MHP utiliza o *set-top box* com mais significância através de um canal de interatividade. Adiciona suporte mandatário para comunicação através de canal de retorno e oferece suporte para conexão IP. Aplicações podem ser recebidas via canal de retorno, ao contrário do *Enhanced Profile* aonde só chegam via *broadcast*;
- c) *Internet Access Profile*: é o perfil destaque do padrão MHP, que objetiva utilização em *set-top boxes* com grande poder de processamento e memória, oferecendo maior suporte a aplicações *web*.

2.3.4.2 Arquitetura

Peng (2002, p. 23), afirma que um modelo da arquitetura MHP consiste de três camadas, descrevendo desta forma elementos de *hardware* e *software*:

- a) camada de recursos: nesta camada estão incluídos os recursos de *hardware* embutidos na televisão ou *set-top box*. Entre os recursos disponíveis incluem *hardware* de decodificação MPEG, dispositivos de entrada e saída de dados, unidade de processamento, memória e sistemas de geração de imagem;
- b) camada de *Software*: as aplicações não acessam diretamente os recursos de *hardware*. A camada de *software* traz uma visão abstrata dos recursos disponíveis. Isolar assim as aplicações do *hardware* permite a portabilidade das aplicações desenvolvidas no ambiente MHP. A camada de software inclui um gerenciador de aplicações, que é uma aplicação residente responsável por controlar o ciclo de vida das aplicações;
- c) camada de Interface das Aplicações: esta camada mantém a interoperabilidade das diversas aplicações MHP desenvolvidas. A DVB-J API é um conjunto de funções de alto nível, estruturas de dados e protocolos que representam uma interface padrão para o desenvolvimento de *software* independente de plataforma de *hardware*.

A organização das três camadas pode ser visualizada na Figura 7.



Fonte: Casas (2006, p. 30).

Figura 7 – Modelo da arquitetura MHP

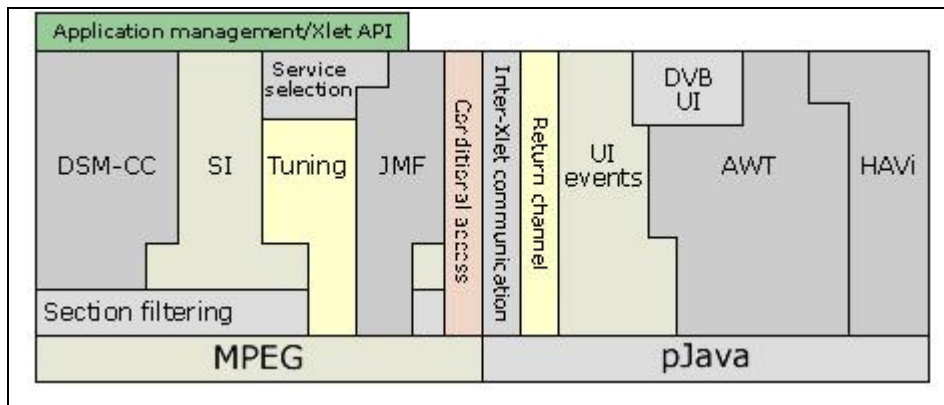
2.3.4.3 Pilha de *software*

Ao visualizar o gerenciamento de aplicações a partir de um nível mais superficial, percebe-se que a API do MHP pode ser dividida em duas partes distintas. De acordo com Morris (2002), uma dessas partes relaciona-se com o fluxo do sinal televisivo por MPEG, carrossel de dados, seleção de serviços e recursos do *Java Media Framework* (JMF), mais direcionada ao *hardware* do *set-top box*, enquanto a outra parte está diretamente ligada com as APIs do Java.

Voltando a atenção para as APIs voltadas à parte do Java, a principal biblioteca a ser citada é a *Abstract Window Toolkit* (AWT), classe que provê recursos de interface para o usuário. Segundo Andreatta (2006), algumas limitações são impostas nas implementações desta biblioteca para o MHP, tendo como objetivo tornar as aplicações mais leves. Entre as principais bibliotecas existentes com implementações de componentes usando como base a AWT são a *DVB User Interface* (DVB-UI), *Home Audio/Video Interoperability* (HAVi) e *Digital Audio Visual Council* (DAVIC). Fernandes et al. (2004) afirmam que além de componentes de interface com o usuário, estas bibliotecas também definem diversas APIs relacionadas a informações de serviços, filtragem de informações, notificação de modificações nos recursos, sintonização de canais de transporte (*tuning*) e controle de acesso.

O diagrama encontrado na Figura 8 apresenta a forma como as APIs se relacionam

conceitualmente entre si.



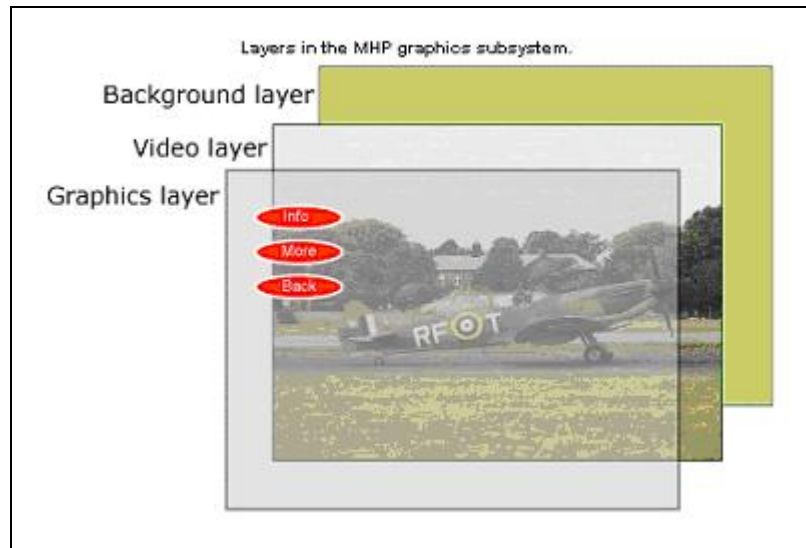
Fonte: Morris (2002).

Figura 8 – Pilha de *software* do MHP

2.3.4.4 Interface gráfica para o usuário

Guimarães (2005, p. 28) define que a *Graphical User Interface* (GUI) API da plataforma MHP inclui funcionalidades para desenhar gráficos (ou partes deles) no dispositivo de saída e para entrada de eventos do dispositivo de entrada. Ela é baseada no pacote de ferramentas AWT com adicionais extensões de TVDI. O pacote `Java.awt` provê um interface de usuário API para permitir aplicações escritas em Java para gerar saída gráfica e receber entrada de eventos de usuário. O AWT foi projetado para fornecer um conjunto comum de ferramentas para projeto de GUI que trabalha numa variedade de plataformas.

Na Figura 9 é demonstrada a representação das camadas da interface gráfica do MHP, que consiste em um plano de fundo, uma camada de vídeo e outra de gráficos.



Fonte: Casas (2006, p. 34).

Figura 9 – Camadas da interface gráfica de uma aplicação MHP

O plano de fundo tem por finalidade definir uma cor de tela quando os outros planos estiverem desabilitados ou não visíveis. Este plano pode ser definido para ter uma cor à escolha do desenvolvedor, para ser um tipo de “tela azul”, ou ter uma imagem fixa, como o logotipo do fabricante. O plano de vídeo, por sua vez, tem por finalidade exibir a saída de vídeo em execução, possuindo propriedades de translação e escalonamento para que possa ser redimensionado em algumas resoluções possíveis e dessa forma transladado para qualquer ponto na tela, de acordo com a necessidade da aplicação. No plano gráfico obtém-se a exibição ao usuário da interface da aplicação. A aplicação pode ter resolução diferente do plano de vídeo, podendo ser desde a resolução mínima de 640x480 *pixels* até a resolução máxima permitida pelo equipamento (PENG apud ANDREATA, 2006, p. 31).

Pozzo (2005, p. 5), enumera as três principais APIs para interface de usuário no MHP: HAVi, DAViC e DVB, que são descritas a seguir.

A interface de usuário definida pela HAVi permite que aplicações escritas em Java determinem os recursos presentes em cada *set-top box* e com isso desenhem sua interface gráfica na tela, manipulem dados entrados pelo usuário, entre outras operações. Ela usa um subconjunto da AWT e estende seus pacotes para suportar recursos específicos da plataforma HAVi.

Baseado no padrão estabelecido pela *Multimedia and Hypermedia Experts Group* (MHEG-6), publicado em 1998, adicionou um conjunto de novas APIs Java, capacitando a essa nova API a acessar informações de serviço, controlar a apresentação de áudio e vídeo e gerenciar recursos do *set-top box*.

A especificação DVB/MHP é construída a partir de outras APIs como JavaTV, HAVi,

e DAVIC. Para completar a especificação DVB/MHP são necessárias novas interfaces de programação que são definidas pela API DVB Core que estende algumas funcionalidades oferecidas por outras APIs como também inclui novas funções para atender aos requisitos da especificação.

2.4 BIBLIOTECAS DE SUPORTE A MIDDLEWARES JAVATV E XLETS

JavaTV (SUN MICROSYSTEMS, 2001) é uma API que estende a plataforma Java e foi desenvolvida pela Sun Microsystems para prover acesso e funcionalidades num receptor de televisão digital. Entre as facilidades providas pela biblioteca, encontram-se o suporte de guias de programação eletrônicas (EPGs), seleção de programas de televisão, gráficos e outros serviços. Contudo, uma das mais importantes facilidades incluídas com JavaTV é a de manipular aplicações Java, que podem ser executadas nos *set-top boxes*.

Um programa Java que executa em um *set-top box* em conformidade com a biblioteca JavaTV recebe o nome de *Xlet*. Cada *set-top box* possui um gerenciador de aplicações, que controla os estados das *Xlets* permitindo iniciar, destruir, pausar e continuar a execução dessas aplicações.

No Quadro 1 é demonstrado o esqueleto de uma *Xlet* simples.

```
import javax.tv.xlet.*;

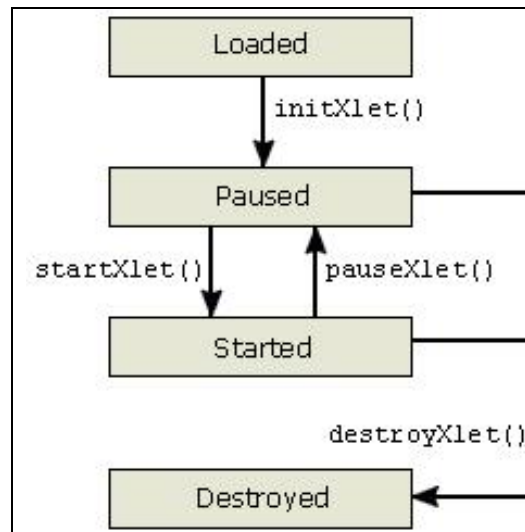
public class FirstXletExample implements javax.tv.xlet.Xlet {
    private javax.tv.xlet.XletContext context;

    public FirstXletExample() {
        // construtor da classe
    }
    public void initXlet(javax.tv.xlet.XletContext context)
        throws javax.tv.xlet.XletStateChangeException {
        // define os parâmetros de inicialização da xlet
        this.context = context;
    }
    public void startXlet()
        throws javax.tv.xlet.XletStateChangeException {
        // inicializa a xlet ou retorna de uma pausa
    }
    public void pauseXlet() {
        // pausa a xlet
    }
    public void destroyXlet(boolean unconditional)
        throws javax.tv.xlet.XletStateChangeException {
        // finaliza a xlet
    }
}
```

Fonte: MHP Multimedia Home Platform (2003).

Quadro 1 – Exemplo de uma *Xlet* e seus principais métodos

De acordo com Montez e Becker (2005, p. 28), “uma *Xlet* em um ambiente de televisão interativa equivale a um *Applet* Java em um *browser* para um computador”. Assim como um *Applet*, uma *Xlet* possui uma interface que permite que uma fonte externa, no caso o gerenciador de aplicação do *set-top box*, inicie-a, pare-a, ou a controle de várias outras formas. O ciclo de vida de uma *Xlet* pode ser visto na Figura 10.



Fonte: Bezerra (2004, p. 12).

Figura 10 – Ciclo de vida de uma *Xlet*

A maior diferença entre um *Applet* e uma *Xlet* é que a segunda pode também ser paralisada e depois ter sua execução continuada. Essa diferença em relação a um *Applet* deve-se ao fato de que, em um ambiente de televisão interativa, várias *Xlets* podem estar sendo executadas simultaneamente, e diferentemente de um computador, um *set-top box* possui menos recursos de hardware. Paralisando uma *Xlet* que se torna invisível, economizam-se valiosos recursos de máquina, tornando possível a execução e exibição de outras aplicações.

2.5 RSS

RSS é uma sub-linguagem baseada no padrão XML, que foi projetada para compartilhar manchetes e outros conteúdos *web*. O projeto foi originado pela *UserLand* em 1997 e subsequentemente utilizado pela empresa Netscape para prover um canal de notícias para um de seus produtos – o *Netcenter*.

Neste formato, cada arquivo de texto com conteúdo RSS, chamado de *feed*, contém informações estáticas sobre o *site* que gerou a informação. Além disso, possui as informações sobre as atualizações e notícias (WEBREFERENCE, 2006).

Casas (2006, p. 22) diz que a principal aplicação deste serviço atualmente é a de manter os usuários de determinadas páginas na Internet que possuem este recurso, para mantê-los informados sobre atualizações desta página sem a necessidade destes usuários acessarem várias vezes o mesmo endereço para saberem das atualizações, fazendo-o apenas quando encontrar alguma atualização que considerar conveniente.

Todo o conteúdo de um *feed* RSS segue o padrão XML, desenvolvido pela entidade chamada W3C (WORLD WIDE WEB CONSORTIUM, 2004), desenhado especialmente para documentos *web* formatando o documento a partir de *tags* que descrevem o conteúdo de cada campo contido no arquivo.

2.5.1 Vantagens de utilização do RSS

Casas (2006, p.22) descreve algumas vantagens da utilização do formato RSS, dentre as principais estão:

- a) grande aceitação pela comunidade de desenvolvedores para a *web*;
- b) rapidez para a elaboração de conteúdo para um arquivo RSS;
- c) padronização das informações por meio do XML, facilitando a elaboração de programas que efetuam o recebimento e leitura de *feeds*;
- d) facilidade de compreensão do arquivo ao abri-lo em um navegador Internet.

2.5.2 Estrutura de um arquivo RSS

De acordo com Dohms (2006), dentro de um *feed* a estrutura do arquivo apresenta um elemento chamado *channel*, que simboliza o canal que está disponibilizando as notícias. Um canal pode conter uma quantidade qualquer de itens, onde cada item representa uma manchete, onde sua descrição é a sinopse ou até mesmo a notícia completa em si.

Casas (2006) ilustra no Quadro 2 os elementos básicos que fazem a identificação do canal que disponibiliza o *feed* RSS e no Quadro 3 alguns elementos opcionais. Estes elementos são importantes para que o programa que faz o gerenciamento dos *feeds* possa manipular de forma específica arquivos recebidos de diferentes canais que provêm o serviço.

Elemento	Descrição	Exemplo
Title	O nome do canal	CNN.com - Tecnologia
Link	O endereço para a página na Internet correspondente ao canal	http://www.cnn.com/rssclick/TECH/?section~cnn_tech
Description	Descrição sobre o tipo de informação disponibilizada pelo canal	CNN.com entrega minuto-a-minuto notícias e informação sobre os últimos acontecimentos, previsão do tempo e muito mais

Fonte: Casas (2006, p. 23).

Quadro 2 – Elementos requeridos para a representação de um canal em um arquivo RSS

Elemento	Descrição	Exemplo
Language	Especificação da linguagem utilizada na informação do RSS. Este elemento deve ser descrito com uma lista de valores internacionais	"br" representa português do Brasil
Copyright	Avaliação da situação de <i>copyright</i> .	Copyright 2005
ManagingEditor	Endereço de <i>e-mail</i> do responsável pelo editorial	editor@cnn.com
Webmaster	Endereço de <i>e-mail</i> do responsável pela publicação do arquivo	webmaster@cnn.com
PubDate	A data de publicação do conteúdo do canal	Sat, 07 Sep 2005 16:00:03 GMT
Lastbuilddate	A data da última vez em que o conteúdo do canal foi modificado	Sat, 07 Sep 2005 09:00:31 GMT
Category	Especifica uma ou mais categorias nas quais enquadram-se as notícias do canal	<category>Newspaper</category>
Generator	Texto que identifica o programa utilizado para gerar o canal	MyghtlynHouse Content System v2.3
Does	O endereço URL que aponta para a documentação para o formato usado no arquivo RSS	http://blogs.law.harvard.edu/tech.rss
Cloud	Permite aos processos registrarem uma sinalização para notificar atualizações do canal	<cloud domain="rpc.sys.com" port="80" path"/RPC2" registerProcechure="pingMe" protocol="soap"/>
Ttl	Quantidade de tempo representada em minutos que indica a duração entre uma atualização e outra da fonte	<ttl>100</ttl>
Image	Indica a forma que é exibida uma imagem no formato GIF, JPEG ou PNG.	
Rating	A qualificação de nível de restrição de acesso para o canal	
TextInput	Especifica uma caixa para entrada de texto que pode ser exibida com o canal	
SkipHours	Uma dica para os programas agregadores de RSS informando as horas em que o canal não deve ser lido	
SkipDays	Uma dica para os programas agregadores de RSS informando os dias da semana em que o canal não deve ser lido.	

Fonte: Casas (2006, p. 23).

Quadro 3 – Elementos opcionais para representação de um canal em um arquivo RSS

O Quadro 4 demonstra os elementos que formam cada item contendo uma notícia, onde os principais elementos-filhos são o título, o endereço para a página onde está a notícia, a descrição e a data de publicação.

Elemento	Descrição	Exemplo
Title	O título do item	Seu computador empoeirado pode conter a cura para o Câncer
Link	O endereço no formato URL do item	http://www.cnn.com/rssclick/2007/TECH/07/05/research.at.home.ap/index.html?section=cnn_tech
Description	A sinopse da notícia	Leia a história completa para maiores detalhes
Author	O endereço de <i>e-mail</i> do autor do item	
Categoria	Inclui o item em uma ou mais categorias	
Comments	O endereço para notícias relacionadas ao item	
Enclosure	Descreve um objeto de mídia que possa estar anexo	
Guid	Texto para identificação única do item	
PubDate	Data em que o item foi publicado	Mon, 05 Jul 2007 10:33:59 EDT
Source	O canal RSS de origem do item	

Fonte: Casas (2006, p. 25).

Quadro 4 – Elementos opcionais para representação de um canal em um arquivo RSS

No Quadro 5 é ilustrado um exemplo de como é organizada a estrutura de um arquivo RSS, onde tem-se a declaração padrão de XML e o elemento raiz RSS. Dentro deste RSS temos o elemento *channel* que representa o canal, onde são subdivididos os itens de acordo com as categorias determinadas como, por exemplo: esportes, política, economia, etc. Dentro de cada canal estão inseridos os elementos *item*, com as matérias.

```
<?xml version="1.0"?>
<rss version="2.0">
<channel>
  <title>Sample Channel</title>
  <link>http://www.somepage.com/</link>
  <description>This is a sample RSS feed</description>
  <copyright>2005</copyright>
  <image>
    <title>Some Page</title>
    <width>88</width>
    <height>31</height>
    <url>http://www.somepage.com/logo.gif</url>
  </image>
  <item>
    <title>This is the first item</title>
    <link>http://www.somepage.com/item1.htm</link>
    <description>Sample item</description>
    <pubDate>Sun, 22 May 2005 22:16:32 GMT</pubDate>
  </item>
</channel>
</rss>
```

Fonte: Dohms (2006).

Quadro 5 – Exemplo de um *feed* RSS

2.6 EMULADORES

Laureano (2005) define um emulador como um software implementado com instruções e funcionalidades existentes em um equipamento real, tendo o objetivo de realizar a simulação virtual deste equipamento. Darós (2004, p. 32) diz que para compreender o funcionamento do conceito da distribuição de dados e sua relação com a disponibilização de serviços interativos em sistemas de TV digital, faz-se necessário um *set-top box* para o desenvolvimento e testes. Mas, em contrapartida, este equipamento ainda não se encontra disponível para comercialização. Visando suprir essa necessidade, existem os ambientes de emulação para reproduzir o funcionamento de TV digital.

Entre os ambientes disponíveis para uso livre na Internet que oferecem suporte a testes de *Xlets* desenvolvidos para a plataforma MHP, apresentam-se o XletView, que pode ser encontrado em Sveden (2004) e o OpenMHP (2003). Ambos são ferramentas *open-source* e apresentam uma implementação parcial da API.

2.6.1 Ambiente de emulação OpenMHP

O OpenMHP (2003) é um projeto *open-source* que tem como objetivo auxiliar no desenvolvimento e testes de aplicações baseadas na especificação DVB-MHP.

A prioridade da equipe de desenvolvimento deste ambiente é a de propiciar aos desenvolvedores de pequenas aplicações, estudantes, entre outras organizações um ambiente que atenda as especificações do padrão MHP, com suporte e implementação das bibliotecas de classes requeridas por esta especificação (DAVIC, DVB e HAVi).

O ambiente de trabalho do OpenMHP é constituído por um painel principal, que simula uma tela de televisão (Figura 11.1). Nesta tela podem ser inseridos como plano de fundo uma imagem estática, a exibição de algum conteúdo em vídeo ou até mesmo a simulação de um carrossel de dados, com os canais de áudio, vídeo e dados. Ao lado desta tela, encontra-se um controle remoto (Figura 11.2), onde são efetuados os comandos de entrada para o *set-top box*, como a troca de canais ou demais funções do aparelho. Por fim, há um painel que exibe informações sobre a execução da *Xlet* (Figura 11.3), como as classes que são carregadas e mensagens de erro emitidas pela aplicação, da mesma forma como um console de um ambiente de desenvolvimento, como por exemplo, o Eclipse (ECLIPSE, 2001).



Figura 11 – Ambiente de emulação OpenMHP

2.7 O SISTEMA BRASILEIRO DE TV DIGITAL

No Brasil, a televisão possui um papel fortemente integrador. Sua presença vem sendo fundamental como agente de cultura, lazer e exercício da cidadania. A televisão brasileira é uma das maiores do mundo - mais de 90% dos domicílios brasileiros possuem televisores e, dentre esses, 80% recebem exclusivamente sinais da televisão aberta (terrestre). É de se esperar que a TV Digital Terrestre venha a ter o mesmo tipo de penetração popular (MINISTÉRIO DAS COMUNICAÇÕES, 2003).

Nesta seção é apresentado o histórico das pesquisas efetuadas no Brasil até chegar na definição de um sistema próprio para TVDI e a seguir são enumerados os principais objetivos da implantação do SBTVD.

2.7.1 Histórico

Montez e Becker (2005, p. 35), citam que os estudos sobre a formação de um sistema brasileiro de TVDI têm raízes na criação da Comissão Assessora para Assuntos de Televisão (COM-TV), estabelecida pelo Ministério das Comunicações em 1991. A comissão tinha como objetivo principal o estudo e a análise da TV de alta definição, em desenvolvimento em alguns países, principalmente no Japão e nos EUA, e em discussão no âmbito da União Internacional de Telecomunicações (UIT). Após o surgimento dos sistemas digitais, a TV de alta definição passou a ser chamada de televisão digital, uma vez que praticamente pararam os estudos sobre a TV analógica.

Segundo Freitas (2004, p. 7), em 1994, a criação do Grupo Técnico Abert/SET de TV Digital marcou o início dos estudos acerca do tema no País. As concessionárias de radiodifusão integrantes do grupo tinham a intenção de acompanhar a evolução da tecnologia e de propor soluções ao governo, representado então pela COM-TV, do Ministério das Comunicações.

Centrando os estudos nos três padrões existentes, o grupo avançou em 1998, quando a Anatel iniciou os seus estudos sobre TV digital e mercado de telecomunicações. Além de tomar a frente nas pesquisas, a Agência avalizou a iniciativa Abert/SET, dando continuidade ao trabalho que vinha sendo desenvolvido, porém com uma visão mais pragmática. O objetivo inicial estava claro: escolher um dos três padrões para ser adotado pelo Brasil. O desenvolvimento de um padrão nacional estava praticamente fora de questão (MONTEZ; BECKER, 2005, p. 35).

De acordo com Freitas (2004, p. 9), em abril de 2001, a ANATEL colocou em consulta pública um relatório, produzido pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), que mudou o foco e o conteúdo do debate. O Relatório Integrador dos Aspectos Técnicos e Mercadológicos da Televisão Digital inovou a pauta ao discutir os modelos de negócio que podem surgir, os modelos de transição, os fatores sociais e culturais que influenciam o processo, entre outros assuntos. Logo em seguida, devido a conflitos de interesses, ocorreu uma cisão na Abert, cujos membros passaram a discordar do ritmo de implantação da tecnologia no país. Além disso, grupos de interesse ligados às indústrias americana e europeia aumentaram a pressão sobre o governo, questionando o resultado dos testes conduzidos pelo Grupo Abert/SET, que apontavam o padrão japonês como o mais apropriado para adoção no Brasil.

Em 2003, o Ministério das Comunicações interrompeu as negociações em torno da adoção imediata de um dos padrões já disponíveis (ATSC, DVB ou ISDB) e decidiu investir em estudos que avaliassem a viabilidade da produção de soluções nacionais de baixo custo e de aplicações voltadas à redução da exclusão digital, direcionando os trabalhos para a criação de um modelo de referência nacional de TV digital terrestre no Brasil e instituindo o Sistema de TV Digital através do Decreto n.º 4.901 de 26 de novembro de 2003. (MINISTÉRIO DAS COMUNICAÇÕES, 2003).

Conforme Montez e Becker (2005, p. 36), para a gestão e execução do SBTVD foram criados três comitês: Comitê de Desenvolvimento, Comitê Consultivo e Grupo Gestor. Ao primeiro, vinculado diretamente à Presidência da República, compete definir as políticas para o desenvolvimento do sistema, incluindo o desenvolvimento tecnológico, a transição, a regulação e o modelo de negócios a ser adotado. O Comitê Consultivo é uma extensão do Comitê de Desenvolvimento, sendo responsável pela proposta de ações e diretrizes fundamentais ao sistema. O Grupo Gestor é responsável pelas ações determinadas pelos dois Comitês, sendo apoiado pela Financiadora de Estudos e Projetos (FINEP) e pelo CPqD.

2.7.2 Objetivos do programa

De acordo com Ministério das Comunicações (2003), para o Brasil, um padrão de TVDI deve contemplar as reais necessidades da sociedade brasileira, tendo em vista o perfil de renda da população e as novas possibilidades abertas através da interatividade. É consenso que a TVDI brasileira possua atributos de baixo custo, robustez na recepção e flexibilidade. Obedecendo essas premissas, as emissoras poderão escolher esquemas de programação e modelos de negócio de acordo com sua conveniência e dos consumidores, interatividade e promoção de novas aplicações à população, proporcionando educação e cultura, contribuindo com a formação de uma sociedade apta a enfrentar os desafios de um mundo onde a informação e o conhecimento são cada vez mais importantes para alcançar o progresso econômico e o bem-estar social.

Entre os principais objetivos estão:

- a) estabelecer e aumentar a rede de competências nacional, promovendo a efetiva integração das pesquisas brasileiras nas áreas de abrangência desse programa;
- b) apresentar solução técnica inovadora, mantendo e aproveitando a compatibilidade com elementos já padronizados no mercado mundial de TVDI;

- c) ser flexível às condições sócio-econômicas do Brasil;
- d) aproveitar o parque nacional instalado de televisores;
- e) permitir uma implantação gradual, minimizando os riscos e os custos para a sociedade, procurando soluções escaláveis e evolutivas, minimizando legados;
- f) ser configurável para potencial adoção por outros países, facilitando exportação;
- g) promover o adensamento da cadeia de valor e de geração de negócios baseados no sistema de TVDI, consolidando os atores envolvidos;
- h) promover soluções industriais que favoreçam a economia de escala;
- i) promover a cultura digital com a TDVI.

2.8 TRABALHOS CORRELATOS

São descritos a seguir três trabalhos correlatos, entre os quais são: Aplicação de comércio via TV digital (DARÓS, 2004); Interpretador de documentos RSS para sistemas de televisão (CASAS, 2006) e o Portal para aplicações colaborativas em TV digital interativa (ANDREATA, 2006).

2.8.1 Aplicação de comércio via TV digital

Darós (2004) desenvolveu uma aplicação de comércio eletrônico sobre um sistema de TV digital (*t-commerce*), demonstrando os conceitos básicos de desenvolvimento de *software* para essa área, baseando-se no padrão JavaTV (SUN MICROSYSTEMS, 2001). Embora apenas tenha sido implementada a interface para o usuário, esse trabalho apresenta o funcionamento de uma aplicação interativa para TV digital baseada no *middleware* MHP. Este trabalho não apresenta implementações referentes à interatividade por conta das restrições apresentadas pelo ambiente de emulação utilizado pelo autor – o XletView – não simula canal de retorno. A Figura 12 demonstra o funcionamento desta aplicação, no momento em que é confirmada a compra do produto.



Fonte: Darós (2004, p. 47).

Figura 12 – Demonstração do aplicativo de comércio eletrônico

2.8.2 Interpretador de documentos RSS para sistemas de televisão

A idéia principal do trabalho de Casas (2006) é a de demonstrar de forma simplificada o processo de traduzir o conteúdo de um arquivo RSS para variáveis dentro do ambiente MHP. O cronograma de desenvolvimento utilizado para a construção do aplicativo foi dividido em três passos. O primeiro deles foi a de estudar os dois componentes principais da aplicação: por um lado, o arquivo RSS e seus padrões, e por outro lado, o estudo sobre o *middleware* MHP, sendo que ao final deste processo a tarefa principal foi a de encontrar uma forma de juntar estes dois elementos para o desenvolvimento da aplicação. O segundo passo foi pesquisar as técnicas de tradução do arquivo XML para Java e de implementação e integração da *Xlet* no ambiente de emulação. Por fim, a terceira e última etapa realizada foi a de obter o entendimento sobre o processo de recepção do conteúdo armazenado em servidores *web* por meio do ambiente de TVDI.

2.8.3 Portal para aplicações colaborativas em TV digital interativa

No trabalho de Andreato (2006), foi desenvolvido um portal para disponibilizar aplicações colaborativas no ambiente de TV digital interativa. O portal foi planejado e implementado através de um protótipo – batizado de InteraTV – mantendo compatibilidade com a camada de software MHP do sistema europeu de televisão digital (DVB). Com o objetivo de utilizar o portal como apoios à área da educação, foram abordadas formas de ensino à distância baseadas em TV digital interativa, conhecido como T-Ensino. Foram apresentadas aplicações educacionais comerciais e acadêmicas de T-Ensino e discutidas suas viabilidades de implementação. Foi descrito o modelo funcional básico do InteraTV, detalhando os seus módulos. Descreveu-se também a simulação do ambiente de TVDI e foram abordadas algumas diferenças entre a forma de desenvolver aplicações para o ambiente da TV digital em comparação com o tradicional ambiente computacional.

3 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo são descritos os requisitos, a especificação, a implementação e a operacionalidade da aplicação desenvolvida. Ao final deste capítulo, os resultados obtidos são comparados com os trabalhos correlatos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação desenvolvida neste trabalho segue os seguintes requisitos:

- a) a aplicação deverá utilizar o canal de retorno para enviar mensagens de controle ao servidor e requisitar o recebimento de *feeds* RSS contendo manchetes de notícias que serão exibidas na tela (Requisito Funcional – RF);
- b) a aplicação deverá possuir um mecanismo de controle de acesso, para que possam ser controlados os canais de serviço que podem ser disponibilizados para cada usuário da aplicação (RF);
- c) a exibição das manchetes poderá ser suspensa e reiniciada, de acordo com os comandos efetuados pelo usuário (RF);
- d) a linguagem adotada para o desenvolvimento do servidor será Java, e o ambiente a ser utilizado será o Eclipse (2001) (Requisito Não-Funcional– RNF);
- e) a especificação da *Xlet* deverá obedecer a especificação encontrada em DVB-MHP (2003) utilizando o emulador OpenMHP para testes e validação (RNF);
- f) a aplicação deverá ter respostas em tempo real e consumir o mínimo de recursos de hardware, levando em conta o fato que uma *Xlet* é executada em um *set top-box*, que possui limitados recursos de máquina (RNF).

3.2 ESPECIFICAÇÃO

Para especificação da aplicação é utilizada orientação a objetos, representada através dos diagramas de casos de uso, de classes e de estados. Para a elaboração destes diagramas, é

utilizada a *Unified Modeling Language* (UML), com as ferramentas *Rational Rose* (IBM CORPORATION, 2006) e *JUDE* (2006). A seguir, são apresentados detalhadamente cada um desses documentos.

3.2.1 Casos de uso

A aplicação possui seis casos de uso, que correspondem às principais funcionalidades da aplicação: efetuar *login*, configurar opções de funcionamento, requisitar o recebimento de notícias, consultar detalhes da notícia, adicionar e excluir notícias aos itens favoritos. Os seis casos de uso estão ilustrados na Figura 13 e a descrição de cada um deles está disponível nos quadros 6, 7, 8, 9, 10 e 11.

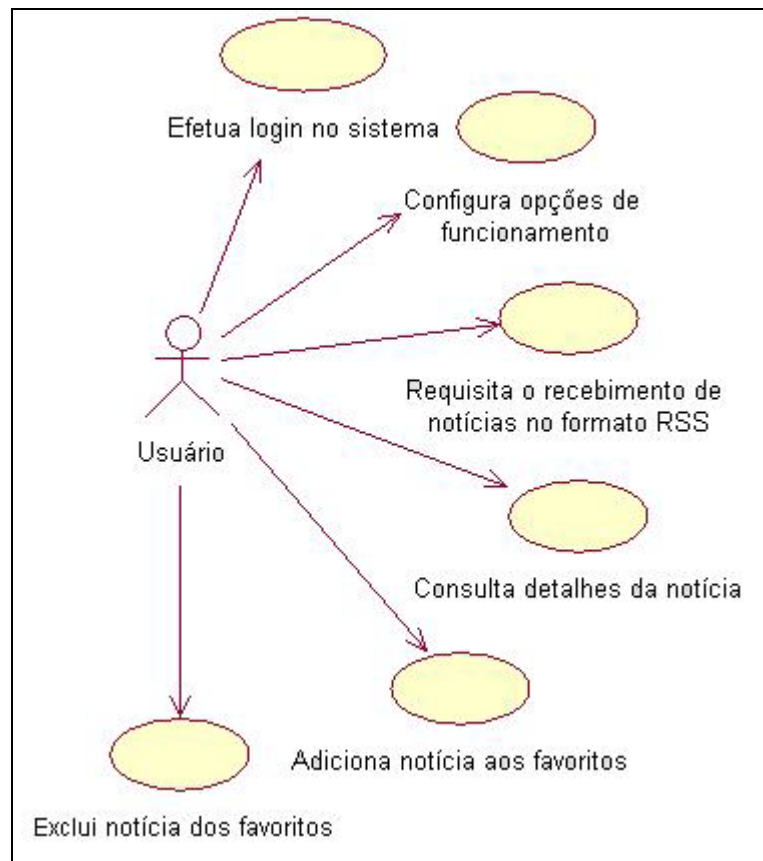
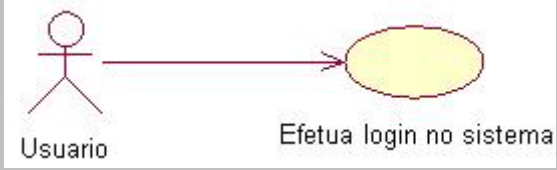
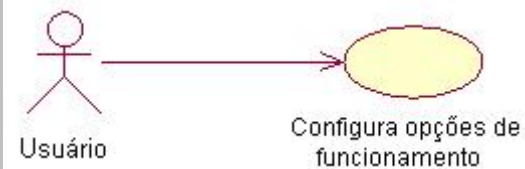



Figura 13 – Diagrama de casos de uso

	
UC. 1. Efetua <i>login</i> no sistema: permite ao usuário acessar o sistema.	
Pré-condições	1. O usuário deve estar cadastrado no sistema.
Fluxo principal	1. O usuário fornece o nome e senha.
Fluxo alternativo	Não possui.
Fluxo de exceção	1. Usuário ou senha inválidos. O sistema informa ao usuário que os dados fornecidos estão incorretos.
Pós-condições e saídas	1. Permitir a entrada do usuário no sistema.

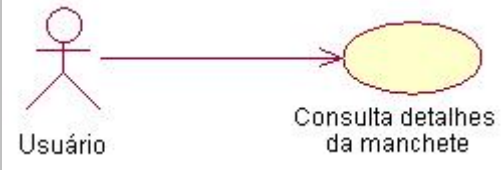
Quadro 6 – Caso de uso: Efetuar *login* no sistema

	
UC. 2. Configura opções de funcionamento: permite ao usuário alterar o tipo de notícias que deseja receber e outras configurações.	
Pré-condições	1. O usuário deve ter efetuado <i>login</i> no sistema. 2. Os canais de notícias disponíveis a este usuário devem estar carregados na lista.
Fluxo principal	1. O usuário seleciona o item que deseja alterar. 2. O usuário escolhe uma das opções disponibilizadas na tela para o item que estiver ajustando.
Fluxo alternativo	Não possui.
Fluxo de exceção	Não possui.
Pós-condições e saídas	1. A aplicação está pronta para efetuar a requisição de notícias ao servidor.

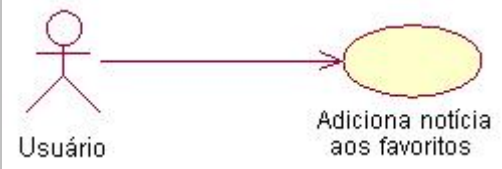
Quadro 7 – Caso de uso: Configurar opções de funcionamento

	
UC. 3. Requisita o recebimento de notícias no formato RSS: permite ao usuário acessar o sistema.	
Pré-condições	1. O usuário deve ter selecionado um canal de notícias.
Fluxo principal	1. O usuário fornece o nome e senha.
Fluxo alternativo	Não possui.
Fluxo de exceção	1. Não conseguiu efetuar conexão com o servidor. O sistema informa ao usuário que o serviço está indisponível.
Pós-condições e saídas	1. Receber o arquivo com as notícias solicitadas.

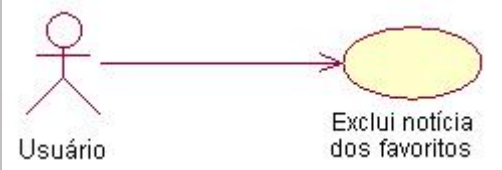
Quadro 8 – Caso de uso: Requisitar o recebimento de notícias no formato RSS

	
<p>UC. 4. Consulta detalhes da manchete: permite ao usuário consultar a descrição detalhada da manchete que está exibida na tela.</p>	
Pré-condições	1. A aplicação deve estar exibindo as manchetes.
Fluxo principal	1. O usuário seleciona a opção “Exibir detalhes” da manchete que está na tela.
Fluxo alternativo	Não possui.
Fluxo de exceção	Não possui.
Pós-condições e saídas	1. A descrição da manchete selecionada pelo usuário é exibida no painel.

Quadro 9 – Caso de uso: Consulta detalhes da manchete

	
<p>UC. 5. Adiciona notícia aos favoritos: permite ao usuário arquivar notícias para que possam ser consultadas posteriormente.</p>	
Pré-condições	1. A aplicação deve estar exibindo as manchetes. 2. O usuário deve ter selecionado uma manchete e aberto o painel com os detalhes desta manchete.
Fluxo principal	1. O usuário seleciona a opção “Adicionar a favoritos”.
Fluxo alternativo	Não possui.
Fluxo de exceção	Não possui.
Pós-condições e saídas	1. A notícia selecionada pelo usuário está armazenada.

Quadro 10 – Caso de uso: Adicionar notícia aos favoritos

	
<p>UC. 6. Exclui notícia dos favoritos: permite ao usuário excluir notícias que estejam gravadas entre os itens favoritos.</p>	
Pré-condições	1. A aplicação deve estar exibindo os itens armazenados nos favoritos. 2. O usuário deve ter selecionado um item para que seja excluído.
Fluxo principal	1. O usuário informa a notícia que deseja excluir. 2. O usuário seleciona a opção “Remover dos favoritos”.
Fluxo alternativo	Não possui.
Fluxo de exceção	Não possui.
Pós-condições e saídas	1. A notícia selecionada pelo usuário está excluída dos itens favoritos.

Quadro 11 – Caso de uso: Excluir notícia dos favoritos

3.2.2 Diagrama de classes

Nesta seção encontra-se a descrição da modelagem das entidades definidas para desenvolver a aplicação.

A classe principal da aplicação é chamada `TVDRSS`, onde é implementada a *Xlet* e os recursos de controle da sua execução. Esta classe é responsável pela criação dos componentes utilizados e a alteração entre os estados da aplicação.

Para a tela inicial da aplicação foi implementada a classe `MenuPrincipal`, onde cada botão deste menu é uma instância de um `MenuItemBox`.

A classe `Login` é a responsável por receber os dados do usuário e sua respectiva senha. Para os campos de usuário e senha, são utilizadas duas caixas de texto, definidas em `TextBoxEditavel`. Para que seja possível efetuar entrada de informações nessas caixas de texto, é definida uma classe para implementação de um `TecladoNumerico`, que agrupa um conjunto de botões definidos em `BotaoTecladoNumerico`.

A tela para efetuar configurações das opções de operação é definida na classe `Configuracao`. Para cada opção desta tela, são utilizadas caixas de seleção que imitam o funcionamento de *combo boxes*, definidas na classe `Selecao`.

A exibição das manchetes é controlada pela classe `Manchetes`. Esta classe faz o gerenciamento do *feed* enviado pelo servidor, armazenando o conteúdo de cada manchete em um `ItemRSS`, que consiste em uma estrutura com os elementos do arquivo RSS. A classe responsável por interpretar o arquivo é chamada `DepositoFeeds`, onde cada um dos itens recebidos é armazenado em um `ArrayList`.

Para a abstração da comunicação com o canal de retorno, é definida uma classe chamada `InterfaceCanalRetorno`, utilizada em `Login` e `Manchetes`.

O painel para exibir detalhes da manchete que foi selecionada é definido na classe `Noticias`. O conteúdo da notícia é exibido por meio de um `TextBox`.

O gerenciamento de notícias arquivadas pelo usuário, contemplando as operações de adicionar, remover e exibir os itens arquivados está modelado na classe denominada `Favoritos`.

As classes implementadas na aplicação encontram-se na Figura 14.

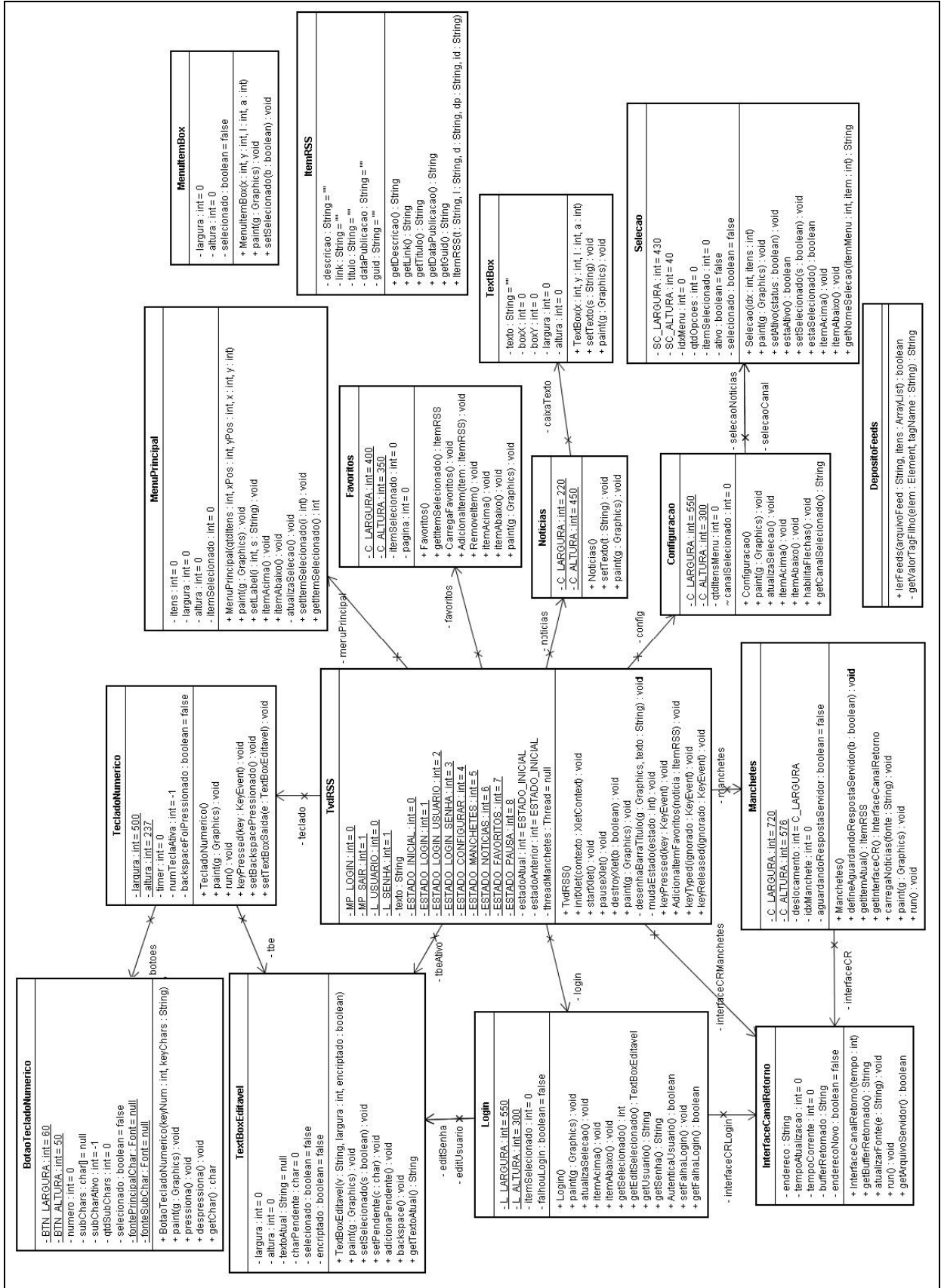


Figura 14 – Diagrama de classes

3.2.3 Diagrama de estados

A Figura 15 contém o diagrama de transição de estados, que ilustra as operações que podem ser efetuadas no decorrer da execução dos processos realizados pela aplicação.

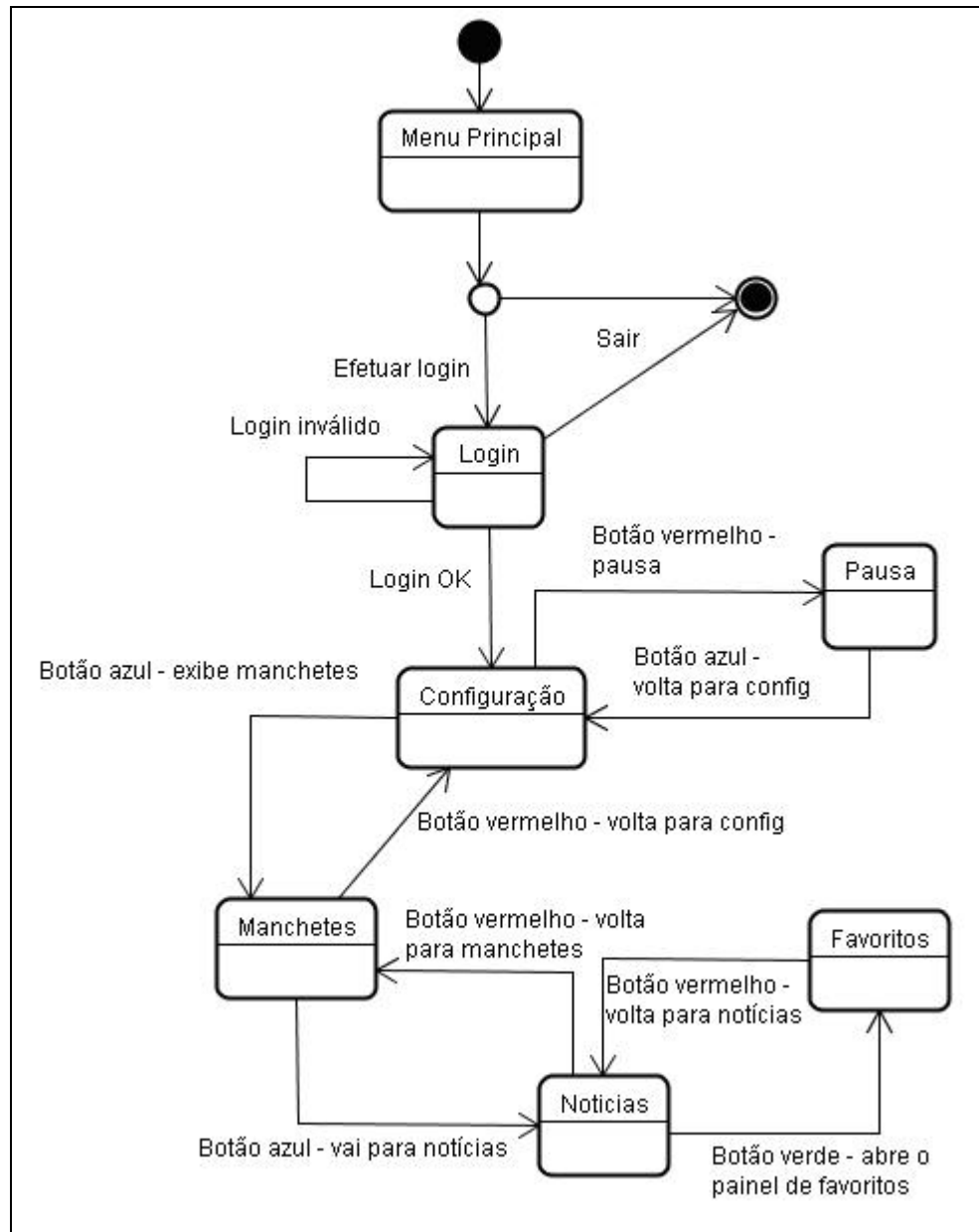


Figura 15 – Diagrama de estados

3.3 IMPLEMENTAÇÃO

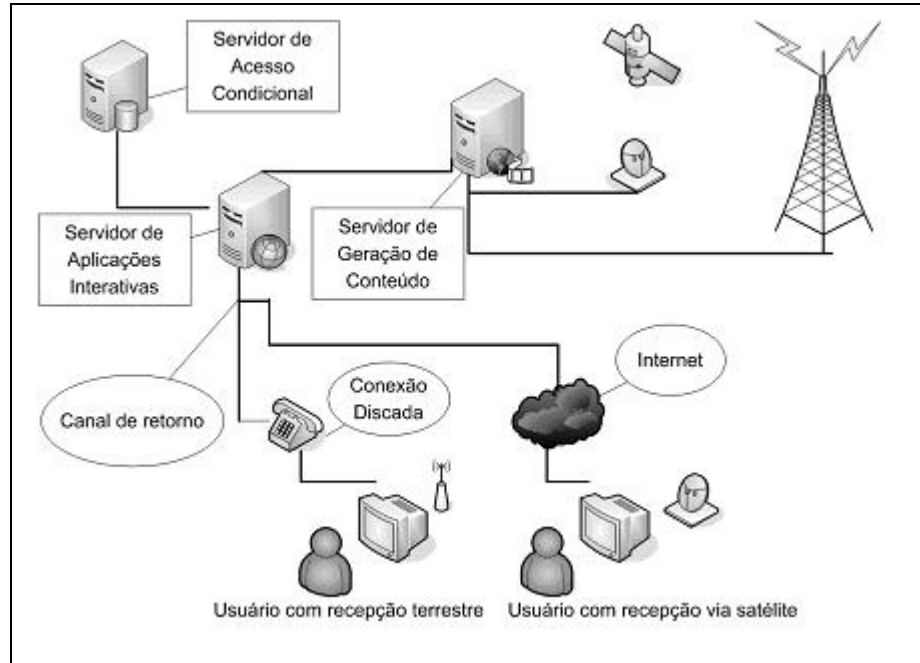
Nesta seção são apresentadas de forma detalhada as técnicas e ferramentas utilizadas no desenvolvimento da aplicação, bem como questões relacionadas à sua operacionalidade.

3.3.1 Técnicas e ferramentas utilizadas

Neste item, são descritos os itens mais relevantes sobre o desenvolvimento da aplicação. Entre os tópicos que recebem destaque estão: o contexto em que a aplicação é executada, a ferramenta utilizada para elaboração do código-fonte, a interface gráfica da aplicação, a conexão com o canal de retorno e a interpretação do arquivo RSS.

3.3.1.1 *Contexto da aplicação*

A aplicação utiliza como base a arquitetura convencional de um sistema de TVDI, como pode ser visto na Figura 16. Especificamente, o *set-top box* deve conter alguma tecnologia de canal de retorno e o usuário deve utilizar-se do controle remoto para efetuar as entradas de informações para a aplicação, eliminando a necessidade de possuir um teclado alfanumérico ou similar. A aplicação foi desenvolvida em definição 4:3, que é o padrão da maioria dos aparelhos de televisão disponíveis na atualidade.



Fonte: Andreatta (2006, p. 61)

Figura 16 - Visão geral do modelo

A interface gráfica tem seu desenvolvimento de forma manual, sem o auxílio de bibliotecas. Sendo assim, para a aplicação são criados componentes para manipulação de caixas de texto, caixas de seleção e botões.

Ao carregar o ambiente de emulação, o gerenciador de aplicações já torna disponível a aplicação para o usuário, que pode começar a interagir com o conteúdo.

A especificação MHP não determina padrões tecnológicos para conexão com canais de retorno, deixando aberta a possibilidade de utilização de tecnologias já conhecidas e estáveis em uso no ambiente da Internet. Sendo assim, a maneira utilizada para conexão com o canal de retorno é feita de forma ponto-a-ponto, onde o usuário submete as requisições de informação e são enviadas ao servidor HTTP, que por sua vez atende as solicitações diretamente ao usuário.

3.3.1.2 Ambiente de desenvolvimento: Eclipse

Desde o início do desenvolvimento da aplicação o ambiente de desenvolvimento escolhido foi o Eclipse (ECLIPSE, 2001) por ser em código aberto, livre, fácil de ser configurado.

Este ambiente de desenvolvimento, mantido e suportado por uma grande comunidade de colaboradores e que possui sempre o seu uso recomendado ainda não possui nenhum tipo

de *plugin* ou adaptação especial ao ambiente de TVDI, mas inicialmente foi utilizado com classes MHP adaptadas ao OpenMHP, conhecidas como *Stub Classes*, por terem sua funcionalidade reduzida para serem compatíveis com ambientes de emulação e posteriormente classes de desenvolvimento disponíveis no ambiente OpenMHP, estas totalmente compatíveis com ambientes reais de TVDI.

3.3.1.3 Interface gráfica e tratamento de eventos

A interface gráfica da aplicação é apresentada na camada gráfica, representada pela classe `HGraphicsDevice`. O elemento principal na construção da interface gráfica é representado pela classe `HScene`, que é semelhante ao `Frame`, utilizados no desenvolvimento de interfaces gráficas Java. Ao criar uma instância de `HScene`, obtém-se uma instância através da classe auxiliar `HSceneFactory`. Uma vez criada a instância de `HScene`, podem ser usados componentes da mesma forma que se usa os componentes AWT do Java.

Essa classe apresenta algumas restrições como, por exemplo, a exibição de uma única instância dessa classe durante qualquer momento da existência da aplicação. Ao finalizar a aplicação é chamado o método `HScene.dispose()` para remover a instância da classe.

O Quadro 12 ilustra a inicialização da *Xlet* e a criação dos componentes visuais utilizados na aplicação.

```
public void initXlet(XletContext context) throws XletStateChangeException {
    this.contexto = context;
    HSceneFactory factory = HSceneFactory.getInstance();
    HSceneTemplate hst = new HSceneTemplate();
    hst.setPreference(HSceneTemplate.SCENE_SCREEN_DIMENSION, new
org.havi.ui.HScreenDimension(1, 1), HSceneTemplate.REQUIRED);
    hst.setPreference(HSceneTemplate.SCENE_SCREEN_LOCATION, new
org.havi.ui.HScreenPoint(0, 0), HSceneTemplate.REQUIRED);

    cena = factory.getBestScene(hst);
    cena.setBounds(0,0,720,576);
    cena.setLayout(null);
    cena.setBackgroundMode(HScene.BACKGROUND_FILL);
    cena.add(this);
    cena.addKeyListener((KeyListener)this);

    scene = HSceneFactory.getInstance().getBestScene(new HSceneTemplate());
    this.setSize(cena.getSize());
    cena.setVisible(false);

    // adiciona os componentes da tela principal
    ...

    // tela de configuração
    config = new Configuracao();
    config.setLocation(110,115);
    config.setVisible(false);
}
```

```

add(config);
...
// define o estado inicial da aplicação
mudaEstado(ESTADO_INICIAL);
}

public void startXlet() throws XletStateChangeException {
    validate();
    cena.setVisible(true);
    ...
    cena.requestFocus();
}

public void destroyXlet(boolean b) {
    if (cena != null) {
        removeKeyListener(this);
        cena.remove(this);
        cena.setVisible(false);
        HSceneFactory.getInstance().dispose(cena);
        cena = null;
    }
    contexto.notifyDestroyed();
}
}

```

Quadro 12 – Tratamento dos eventos do controle remoto

São os seguintes passos utilizados no desenvolvimento interface gráfica de usuário:

- a) criação e instanciação de um objeto `HScreen` para obter os correspondentes objetos das camadas de fundo, de vídeo e gráfica;
- b) instanciação de um objeto `HScene` no contexto da camada gráfica. Nesse objeto são colocados os Contêineres e demais componentes de interface gráfica;
- c) tratamento de eventos relacionados aos componentes da interface gráfica.

As entradas de informações por meio do controle remoto são tratadas como eventos `KeyEvent`, onde o mapeamento de cada botão do controle remoto corresponde às definições encontradas no ambiente de emulação. No Quadro 13 é apresentado um trecho da rotina que efetua o tratamento das entradas por meio do controle remoto, tanto para navegação como para a troca de estados da aplicação.

```

// manipula o comportamento dos menus de acordo com os comandos efetuados
public void keyPressed (KeyEvent key) {
    int comando = key.getKeyCode();
    switch (estadoAtual) {
        ...
        // tratamentos para a tela de login no sistema
        case ESTADO_LOGIN:
            switch (comando) {
                case KeyEvent.VK_UP:      login.itemAcima();           break;
                case KeyEvent.VK_DOWN:    login.itemAbaixo();        break;
                case KeyEvent.VK_ENTER:
                    switch (login.getSelecionado()) {
                        case L_USUARIO:
                            mudaEstado(ESTADO_LOGIN_USUARIO);
                            tbeAtivo = login.getEditSelecionado();
                            tbeAtivo.setSelected(true);
                            teclado.setTextBoxSaida(tbeAtivo);
                            break;
                        case L_SENHA:
                            mudaEstado(ESTADO_LOGIN_SENHA);
                            tbeAtivo = login.getEditSelecionado();
                    }
            }
    }
}

```



```

        tbeAtivo.setSelected(true);
        teclado.setTextBoxSaida(tbeAtivo);
        break;
    }
    break;
    case HRcEvent.VK_COLORED_KEY_0:
        texto = null;
        destroyXlet(true);
        break;
    case HRcEvent.VK_COLORED_KEY_3:
        if (login.getFalhaLogin()) {
            login.setFalhaLogin();
            repaint();
        }
        else
            // usuário ou senha zerados
            if (login.getUsuario().length() == 0 ||
                login.getSenha().length() == 0) {
                login.setFalhaLogin();
                repaint();
            }
        else { // aprova o login
            ...
        }
        if (login.AutenticaUsuario()) {
            texto = null;
            mudaEstado(ESTADO_CONFIGURAR);
        }
        else {
            login.setFalhaLogin();
            repaint();
        }
    }
    break;
    default:    break;
}
break;
// tratamentos ao digitar nos campos da tela de login
case ESTADO_LOGIN_USUARIO:
case ESTADO_LOGIN_SENHA:
    if ((comando >= KeyEvent.VK_0) && (comando <= KeyEvent.VK_9)){
        teclado.keyPressed(key);
    }
    if (comando == KeyEvent.VK_ENTER) {
        tbeAtivo = null;
        mudaEstado(ESTADO_LOGIN);
    }
    if ((comando==HRcEvent.VK_F1) || (comando==HRcEvent.VK_COLORED_KEY_0)) {
        teclado.setBackspacePressionado();
    }
    break;
    default:    break;
}
}
}

```

Quadro 13 – Tratamento dos eventos do controle remoto

Para efetuar *login* no sistema, é necessário que o telespectador informe o nome do usuário e a senha. Para efetuar essa operação, faz-se necessária a implementação de um teclado que utilize a técnica de entrada de texto semelhante à utilizada nos telefones, cuja parte da implementação pode ser vista no Quadro 14. Esse teclado utiliza os botões numéricos de 0 a 9, onde cada tecla numérica também representa três ou quatro letras. Para digitar uma letra, pressionando a tecla que a contém repetidas vezes até que a letra seja exibida, por

exemplo: para digitar a letra “c”, deve ser pressionada a tecla “2” três vezes rapidamente.

```

public void run() {
    while (true) {
        try { Thread.sleep(100); } catch (Exception e) { System.out.println(e); }
        // nt é compartilhada por duas threads,
        // mas somente uma delas pode acessar de cada vez
        synchronized (nt) {
            if (nt.get() != -1) {
                if (numTeclaAtiva == -1) {
                    // nova tecla foi pressionada
                    numTeclaAtiva = nt.get();
                    botoes[numTeclaAtiva].pressiona();
                    tbe.setPendente(botoes[numTeclaAtiva].getChar());
                    nt.set(-1);
                }
                else {
                    if (numTeclaAtiva == nt.get()) {
                        // a mesma tecla foi pressionada novamente
                        botoes[numTeclaAtiva].pressiona();
                        tbe.setPendente(
                            botoes[numTeclaAtiva].getChar());
                        nt.set(-1);
                    }
                    else { // nova tecla apertada enquanto a antiga está ativa
                        tbe.adicionaPendente();
                        botoes[numTeclaAtiva].despressiona();
                        numTeclaAtiva = nt.get();
                        botoes[numTeclaAtiva].pressiona();
                        tbe.setPendente(botoes[numTeclaAtiva].getChar());
                        nt.set(-1);
                    }
                }
                // cada tecla pressionada define o timer como 10
                timer = 10;
            } else { // nenhuma tecla pressionada, decrementa o timer
                timer--;
            }
            if (timer == 0) { // quando o timer alcança 0
                // adiciona a tecla ativada para a string na caixa de texto
                tbe.adicionaPendente();
                botoes[numTeclaAtiva].despressiona();
                numTeclaAtiva = -1;
            }
            // o backspace desativa qualquer caractere que estiver ativo
            if (backspaceFoiPressionado) {
                tbe.backspace();
                if (numTeclaAtiva != -1) {
                    botoes[numTeclaAtiva].despressiona();
                    numTeclaAtiva = -1;
                }
                timer = -1;
                backspaceFoiPressionado = false;
            }
        }
    }
}

// reconhece um botão pressionado do controle remoto
public void keyPressed (KeyEvent key) {
    int botao = key.getKeyCode();
    if ((botao >= KeyEvent.VK_0) && (botao <= KeyEvent.VK_9)) {
        // nt pode ser modificada pelo método run()
        synchronized (nt) {
            nt.set(botao - KeyEvent.VK_0);
        }
    }
}
}

```

Quadro 14 – Implementação de entrada de texto no teclado numérico

3.3.1.4 O canal de retorno

Através do canal de retorno, que na aplicação é implementada na classe `InterfaceCanalRetorno`, ocorrem o envio das informações de *login* do usuário no serviço e as requisições das notícias por meio dos *feeds*. Existe uma abstração do canal de retorno, substituído por conexões TCP/IP, onde as requisições são realizadas através de uma instância da classe `URLConnection`. O Quadro 15 descreve como foi implementada a comunicação com o servidor por meio do canal de retorno.

```

public boolean getArquivoServidor () {
    BufferedReader reader = null;
    try {
        // pega referência para RCInterfaceManager
        RCInterfaceManager rcm = RCInterfaceManager.getInstance();
        // pega a lista de interfaces para o canal de retorno
        // que estão disponíveis para a aplicação
        RCInterface[] interfaces = rcm.getInterfaces();
        // escolhe o tipo de interface para fazer a comunicação
        // verifica se a interface é do tipo ConnectionRCInterface
        if (interfaces[0] instanceof ConnectionRCInterface) {
            // se for do tipo ConnectionRCInterface, não é uma conexão
            // permanente, então precisamos primeiro conectar
            ConnectionRCInterface myInterface;
            myInterface = (ConnectionRCInterface)interfaces[0];
            // agora que há uma referência para a interface
            // pode começar a usá-la
            try {
                // primeiramente, reservamos a conexão
                myInterface.reserve();
                // configura os parâmetros de conexão
                ConnectionParameters myConnectionParameters;
                myConnectionParameters = new ConnectionParameters
                    ("0199604235697", "username", "password");
                // define o destino da conexão
                myInterface.setTarget(myConnectionParameters);

                // efetua a conexão com o servidor
                myInterface.connect();

                // efetua as operações que deseja fazer
                // cria uma conexão para requisitar o arquivo
                HttpURLConnection conexao = (HttpURLConnection) new URL
                    (endereco).openConnection();

                // cria um buffer de leitura para a conexão
                reader = new BufferedReader(new
                    InputStreamReader(conexao.getInputStream()));

                // desconecta a interface e libera o recurso
                myInterface.disconnect();
                myInterface.release();
            } catch (permissionDeniedException e) {
                // não possui permissão para efetuar a operação
                return;
            }
        }
    }
}

```

Quadro 15 – Conexão com canal de retorno

O pacote `org.dvb.net.rc` possui as classes necessárias para o gerenciamento das seções de conexão. Neste pacote destacam-se as classes `RCInterface` que representa uma

interface para o canal de retorno, possuindo métodos para identificar o tipo de canal de retorno existente e velocidade de transferência permitida. A classe `ConnectionRCInterface` representa a interface para um canal de retorno que não é permanentemente conectado, possuindo métodos para gerenciar a reserva do recurso, efetuar a conexão e desconexão, como também lidar com os parâmetros de conexão como número de telefone, usuário e senha. A especificação MHP não diz como deve ser o servidor destas conexões do canal de retorno, deixando então aberto para uso de servidores específicos da própria rede de TV ou de empresas afiliadas.

3.3.1.5 Interpretação do arquivo XML que contém o *feed*

Após receber o arquivo XML do servidor, cuja estrutura já foi apresentada no item 2.5.2 da fundamentação teórica, o conteúdo desse arquivo é mapeado para objetos através da classe `DepositoFeeds`, cujo código é ilustrado no Quadro 15.

```
public boolean lerFeeds(String arquivoFeed, ArrayList itens) throws Exception {
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document doc = db.parse(new ByteArrayInputStream(arquivoFeed.getBytes()));
    Element elem = doc.getDocumentElement();
    // pega todos os elementos item do XML
    NodeList nl = elem.getElementsByTagName("item");
    // percorre cada elemento "item" encontrado
    for (int i = 0; i < nl.getLength(); i++) {
        Element tagItem = (Element)nl.item(i);
        // pega os dados de cada item
        String descr = getValorTagFilho(tagItem, "description");
        String link = getValorTagFilho(tagItem, "link");
        String titulo = getValorTagFilho(tagItem, "title");
        String dataPub = getValorTagFilho(tagItem, "pubDate");
        String guid = getValorTagFilho(tagItem, "guid");
        itens.add(new ItemRSS(titulo, link, descr, dataPub, guid));
    }
    return true;
}
// este método lê e retorna o conteúdo (texto) de uma tag (elemento)
// filho da tag informada como parâmetro. A tag filho a ser pesquisada
// é a tag informada pelo nome (string)
private String getValorTagFilho (Element elem, String tagName) throws Exception {
    NodeList filhos = elem.getElementsByTagName(tagName);
    if (filhos == null) return null;
    Element filho = (Element) filhos.item(0);
    if (filho == null) return null;
    return filho.getFirstChild().getNodeValue();
}
```

Quadro 16 – Interpretação do arquivo XML

São duas as principais APIs para interpretação de XML no Java. Uma delas é o *Simple API for XML* (SAX) e a outra é o *Document Object Model* (DOM). Para a interpretação do arquivo XML na aplicação é utilizado o DOM. Essa API tem como vantagem mapear diretamente um XML na memória, mas em contrapartida o gasto de memória é alto, não

aconselhado para arquivos muito grandes.

Para cada *tag* denominada *item* encontrada no XML é criada uma instância da classe `ItemRSS`, alimentando esta instância com os valores lidos do XML e ao final adiciona esta instância em um conjunto de objetos do tipo `ArrayList`, que compõe a lista de notícias que serão exibidas para o usuário.

3.3.2 Operacionalidade da aplicação

Para demonstrar na prática o funcionamento de uma aplicação interativa para TVDI, foi criado o Serviço Interativo de Notícias, que seja de fácil utilização e tenha comportamento relativamente discreto ao exibir as notícias aos telespectadores.

Atualmente já existem serviços de notícias na televisão que exibem notícias correndo horizontalmente, tanto na parte superior como na parte inferior na tela durante a exibição de um programa televisivo. A aplicação desenvolvida encontra-se em um cenário no qual o telespectador é assinante de um canal de televisão que disponibiliza um serviço de notícias personalizado, onde o telespectador pode optar por receber apenas notícias que sejam do seu interesse.

Ao carregar a aplicação no *set-top box*, é exibida uma tela inicial, como pode ser visto na Figura 17, onde o telespectador pode decidir entre inicializar ou encerrar a aplicação.



Figura 17 – Tela inicial da aplicação

Ao escolher a opção de inicializar o serviço, o telespectador deve efetuar *login* neste serviço (Figura 18). Este recurso tem como objetivo restringir o acesso apenas para assinantes que estejam cadastrados no serviço disponibilizado pelo canal de televisão. Para cada usuário cadastrado no sistema são disponibilizados diferentes canais de notícias de acordo com os assuntos de interesse ou restrições relacionadas a conteúdo inapropriado. Para informar o nome do usuário e senha, o usuário seleciona uma das caixas de texto e aperta o botão OK, que disponibiliza a entrada das informações por meio dos botões numéricos do controle remoto, que assume comportamento semelhante a um teclado de telefone celular. Para encerrar a digitação nos campos, deve ser apertado novamente o botão OK.



Figura 18 – Tela de *login*

Ao terminar de informar os dados para *login*, o usuário confirma por meio do botão azul. No caso de alguma das informações (usuário ou senha) forem informadas de forma incorreta, é exibida uma mensagem de erro informando sobre a existência de erro nos dados informados nesta tela, de forma que o telespectador tenha que corrigi-las para conseguir efetuar o *login*.



Figura 19 – Login inválido

No momento em que o telespectador consegue acessar o serviço, são carregadas as configurações personalizadas para este usuário, como os canais de notícias que lhe são disponibilizadas, e também outras informações, como as notícias catalogadas por este usuário nos favoritos. No painel de configurações (Figura 20), o telespectador pode configurar de qual dos canais deseja receber as manchetes, habilitar ou desabilitar as funções de estender o painel com os detalhes da manchete, os favoritos e a frequência com que o serviço fará a busca por atualizações disponibilizadas pelo servidor. A navegação nesta tela se faz por meio dos botões para cima e para baixo do controle remoto, e para habilitar a alteração do item selecionado em cada uma das opções, o usuário deve pressionar o botão OK e escolher entre as opções utilizando novamente os botões de navegação para cima e para baixo.



Figura 20 – Painel de configurações

O telespectador por meio do botão azul confirma as configurações, fazendo com que a aplicação efetue a requisição das notícias ao servidor. Feito este processo, a aplicação começa a exibição das manchetes na tela, como pode ser visto na Figura 21. Enquanto a aplicação estiver ativa, periodicamente serão requisitadas atualizações das notícias ao servidor. A data e hora da última atualização disponibilizada pelo serviço encontram-se na barra exibida na parte inferior da tela. A qualquer momento, o telespectador pode voltar a acessar a tela de configurações apertando o botão vermelho do controle remoto, podendo suspender temporariamente a exibição da barra de notícias quando considerar conveniente, fazendo com que não seja mostrada nenhuma tela da aplicação.



Figura 21 – Tela de exibição das manchetes

No momento em que o telespectador estiver visualizando uma determinada manchete e quiser saber mais detalhes sobre a notícia, ele pode acessar esta informação apertando o botão verde, que efetua a chamada do painel com a descrição da notícia selecionada, localizado à direita da tela da televisão (Figura 22). A partir deste painel, caso o telespectador desejar arquivar a notícia selecionada, pode adicioná-la aos favoritos.



Figura 22 – Painel de exibição do resumo da notícia

Ao apertar o botão azul no painel que exhibe os detalhes da notícia, o telespectador pode acessar a tela para consultar os itens gravados nos favoritos (Figura 23). Nesta tela, os itens são exibidos com data de geração da notícia e o título, onde o telespectador pode

selecionar cada um desses itens e exibir a descrição detalhada no painel à direita apertando o botão verde ou removê-lo da lista por meio do botão vermelho.



Figura 23 – Tela com os itens gravados nos favoritos

3.4 RESULTADOS E DISCUSSÃO

O objetivo deste trabalho está em demonstrar o desenvolvimento de uma aplicação interativa para o ambiente de TVDI, buscando seguir o padrão tecnológico estabelecido no padrão MHP, sem a adição de novas soluções e visando simular o funcionamento do modelo o mais próximo possível de um ambiente real. Havia a possibilidade de usar apenas linguagem declarativa para a criação da aplicação, mas o uso de linguagem procedural, como o Java, permite a criação de aplicações mais robustas e com maiores funcionalidades, além de uma interface gráfica personalizada e com diversos recursos gráficos pré-existentes.

A aplicação de comércio eletrônico para o ambiente de TVDI desenvolvido por Darós (2004) tinha como meta principal apresentar os conceitos do funcionamento de um sistema de TV digital e a tecnologia existente até então, oferecendo um bom suporte sobre a idéia de como desenvolver aplicações para este ambiente.

Casas (2006) direcionou os esforços de sua pesquisa para interpretar arquivos RSS e inseri-los no contexto de uma aplicação para TVDI no padrão MHP. O propósito desse trabalho, de acordo com as palavras do autor era a de apenas desenvolver o módulo para

recepção e tradução do arquivo, sem se preocupar em desenvolver uma aplicação completa para que pudessem ser demonstradas as funcionalidades.

Andreatta (2006) descreve a criação de um portal de aplicações, com vários módulos interativos, entre eles *chat* e tele-enquete, tendo o compromisso de utilizar o recurso do canal de retorno de forma que o projeto se enquadrasse em um ambiente real de TVDI, sem que o protótipo transformasse o receptor de TV em uma estação de acesso à Internet. Entre os trabalhos correlatos apresentou-se como ótima base conceitual.

4 CONCLUSÕES

Ao final deste trabalho, os resultados obtidos podem ser analisados por vários aspectos conforme são apresentados a seguir.

De forma geral, o trabalho atingiu o objetivo de demonstrar o funcionamento de uma aplicação interativa para TV digital de forma simulada, com o usuário tendo a possibilidade de requisitar informações diretamente a um servidor por meio do canal de retorno. No contexto atual, a idéia de utilizar o canal de retorno para comunicação entre telespectador e o serviço difusor ainda encontra-se em fase de estudos relacionados a viabilidade de implantação e garantir a segurança das informações.

Apesar de haverem vários estudos sobre a questão do desenvolvimento de aplicações que explorem a interatividade total no sistema de TVDI, existem ainda algumas lacunas relacionadas a este item que impedem a sua implementação no sistema brasileiro. Dentre algumas delas podem-se enumerar a forma como serão disponibilizados os recursos interativos, levando em conta fatores socioeconômicos, relacionados diretamente à infraestrutura de cada região para que este recurso não fique restrito a uma quantidade reduzida de telespectadores. Outra possibilidade que vem sendo estudada é a de disponibilizar serviços interativos da mesma forma como se o usuário estivesse acessando recursos via Internet.

No que diz respeito a padrões de desenvolvimento, considerando a linha de tempo percorrida entre o início e a conclusão do trabalho, alguns conceitos se consolidaram em relação à implantação do sistema de TV digital no Brasil, como a opção por desenvolver um middleware próprio – o GINGA – combinando os melhores recursos entre os demais padrões existentes no mundo. A escolha por realizar este trabalho utilizando a plataforma MHP deveu-se principalmente pelo fato do sistema brasileiro ainda ser muito recente e não encontrar-se disponível em nível de documentação e suporte ao desenvolvimento de aplicações. Sendo assim, abre-se a possibilidade de utilizar o padrão brasileiro para futuros trabalhos nesta área.

O emulador OpenMHP mostrou-se satisfatório em relação ao que a ferramenta propõe-se a fazer: disponibilizar os recursos implementados para o padrão MHP ao simular o funcionamento de uma TV digital. Em contrapartida, neste ambiente existem algumas limitações, comuns a outras ferramentas *open-source* disponíveis – nas quais enquadram-se o XletView e o iTV-Espial. – como, por exemplo, a adaptação de algumas classes faz com que alguns aspectos funcionais tenham um comportamento superficial.

4.1 EXTENSÕES

Considerando que o escopo principal deste trabalho foi demonstrar a possibilidade de comunicação do telespectador com o provedor de serviços televisivos, outros aspectos foram desconsiderados no desenvolvimento da aplicação, dentre eles podem ser enumerados:

- a) aprimorar questões relacionadas à usabilidade e interface com o usuário;
- b) explorar recursos de segurança a nível local na aplicação, para que um usuário não tenha acesso a configurações pessoais de outros usuários, como também em relação à proteção dos dados que trafegam pelo canal de retorno;
- c) implementar um interpretador de conteúdo HTML, de modo que o usuário possa acessar a página referente às manchetes que são exibidas na aplicação.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALECRIM, Emerson. **Tecnologia RSS**. [S.l.], 2005. Disponível em: <<http://www.infowester.com/rss.php>>. Acesso em: 11 out. 2006.
- ANDRADE, Fernando C. **Entendendo data broadcasting em plataformas de TV digital**. 2004. 30 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2003-2/fcan.rar>>. Acesso em: 02 nov. 2007.
- ANDREATA, Jomar. **InteraTV: um portal de aplicações colaborativas em TV digital interativa utilizando a plataforma MHP**. 2006. 96 f. Dissertação (Mestrado em Engenharia Elétrica) – Curso de pós-graduação em Engenharia Elétrica – Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <<http://www.das.ufsc.br/~montez/2006%20Jomar%20Andreata.pdf>>. Acesso em: 17 set. 2007.
- BECKER, Valdecir et al. **Júri virtual: uma aplicação para TV digital interativa baseada em JavaTV e HyperProp**. Florianópolis, 2004. Disponível em: <<http://www.i2tv.ufsc.br/docs/Becker%20Webmidia2004.pdf>>. Acesso em: 23 ago. 2006.
- BEZERRA, Edmo S. R. **Estudo dos middlewares MHP e DASE para TV digital**. 2004. 52 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2003-2/esrb.zip>>. Acesso em: 22 ago. 2006.
- CASAS, Ignasi C. **Parsing from RSS feeds to MHP-television systems**. 2006. 46 f. Trabalho de Conclusão de Curso (Engenharia Técnica Industrial) – Universitat Politècnica de Catalunya, Espanha. Disponível em: <<https://upcommons.upc.edu/pfc/bitstream/2099.1/3857/1/54957-1.pdf>>. Acesso em: 22 out. 2007.
- COELHO, Andrino S. S. **Uma especificação de desenvolvimento de serviços para televisão digital interativa**. 2005. 81 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2005-1/assc.pdf>>. Acesso em: 23 ago. 2006.
- DARÓS, Joel A. **Interfaces de desenvolvimento de aplicações para TV digital baseado no middleware MHP**. 2004. 53 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- DOHMS, Rafael. **Um estudo em RSS parte 2: o formato RSS**. Porto Alegre, 2006. Disponível em: <<http://www.rafaeldohms.com.br/2006/07/21/um-estudo-em-rss-parte-2-o-formato-rss/pt/>>. Acesso em: 20 out. 2007.

ECLIPSE. **Eclipse**: an open development platform. [S. l.], 2001. Disponível em: <<http://www.eclipse.org>>. Acesso em: 22 out. 2007.

FERNANDES, Jorge et al. **Introdução à televisão digital interativa**: arquitetura, protocolos, padrões e práticas. Brasília, 2004. 56 f. Disponível em: <http://www.cic.unb.br/~jhcf/MyBooks/itvdi/slides-jai2004/IntroducaoATelevisaoDigitalInterativa_dia1.pdf>. Acesso em: 22 out. 2007.

FREITAS, Igor V. B. **Televisão digital**: que imagem terá o modelo brasileiro. Brasília, 2004. 64 f. Disponível em: <http://www.senado.gov.br/conleg/textos_discussao/texto18%20-%20igor.pdf>. Acesso em: 15 out. 2006.

GUIMARÃES, Rangner F. **Interação do espectador com TV digital**. 2005. 52 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2004-2/rfg.doc>>. Acesso em: 22 out. 2007.

IBM CORPORATION. **Rational Rose data modeler**. [S.l.], 2006. Disponível em: <http://www-111.ibm.com/ecatalog/Detail.wss?locale=pt_BR&synkey=H106683S95271V02>. Acesso em: 21 out. 2007.

JUDE. **Jude**: system design tool. Tóquio, Japão, 2006. Disponível em: <<https://jude.change-vision.com>>. Acesso em: 21 out. 2007.

LAUREANO, Marcos A. P. **Máquinas virtuais e emuladores**. Curitiba, 2005. Disponível em: <www.laureano.eti.br/ensino/fae/aula-05-maquinas-virtuais_emuladores.pdf>. Acesso em: 02 nov. 2006.

MHP MULTIMEDIA HOME PLATFORM. **Digital video broadcasting multimedia home platform**. [S.l.], 2003. Disponível em: <<http://www.mhp.org>>. Acesso em: 23 ago. 2006.

MINISTÉRIO DAS COMUNICAÇÕES. **Sistema brasileiro de TV digital**. Brasília, 2003. Disponível em: <<http://sbtvd.cpqd.com.br>>. Acesso em: 03 nov. 2007.

MONTEIRO, Monique L. B. **Uma proposta de categorização para aplicações de TV digital**. 2004. 51 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2003-2/mlbm.doc>>. Acesso em: 22 ago. 2006.

MONTEZ, Carlos; BECKER, Valdecir. **TV digital interativa**: conceitos, desafios e perspectivas para o Brasil. 2. ed. Florianópolis: Ed. da UFSC, 2005.

MORRIS, Steven. **The interactive TV web**. [S.l.], [2002?]. Disponível em: <<http://www.mhp-interactive.org>>. Acesso em: 07 out. 2006.

OPENMHP. **Free open source MHP compliant library**. [S.l.], 2003. Disponível em: <<http://www.openmhp.org/index.php>>. Acesso em: 19 out. 2006.

PENG, C. **Digital television applications**. 2002. 117 f. Tese (Doutorado em Telecommunications Software and Multimedia Laboratory) – Departamento de Ciências da Computação e Engenharia, Helsinki University of Technology, Finlândia. Disponível em: <<http://lib.tkk.fi/Diss/2002/isbn9512261723/isbn9512261723.pdf>>. Acesso em: 17 ago. 2007.

PICCIONI, Carlos A.; MONTEZ, Carlos. **Um estudo sobre emuladores de aplicações para a televisão digital interativa**. Florianópolis, 2004. Disponível em: <inf.unisul.br/~ines/workcomp/cd/pdfs/2507.pdf>. Acesso em: 25 ago. 2006.

POZZO, Douglas D. **Aplicativos para TV digital interativa**. Florianópolis, [2005?]. Disponível em: <http://www.enapet.ufsc.br/anais/APLICATIVOS_PARA_TELEVISAO_DIGITAL_INTERATIVA.pdf>. Acesso em: 14 ago. 2007.

SILVA, Robert. **Surround sound: the history and basics of surround sound**. [S.l.], 2006. Disponível em: <<http://hometheater.about.com/od/beforeyoubuy/a/surroundsound.htm>>. Acesso em: 26 out. 2006.

SUN MICROSYSTEMS. **JavaTV API specification**. [S.l.], 2001. Disponível em: <<http://java.sun.com/products/javaTV>>. Acesso em: 07 out. 2006.

SVEDEN, Martin. **Xletview sourceforge project**. [S.l.], 2004. Disponível em: <<http://xletview.sourceforge.net>>. Acesso em: 12 out. 2007.

TAKEGUMA, Itapajé F. S. **Um modelo para distribuição de conteúdo visual em um ambiente de TV digital**. 2005. 33 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2005-1/monografia.doc>>. Acesso em: 22 ago. 2006.

UNICAMP. **Inovação Unicamp**. Campinas, 2006. Disponível em: <<http://www.inovacao.unicamp.br/report/news-tvdigital060807.shtml>>. Acesso em: 02 nov. 2006.

VIDEOLAR. **Videolar S.A.** [S.l.], [2002?]. Disponível em <<http://www.videolar.com.br/paginas/faq.asp>>. Acesso em: 02 nov. 2006.

VIEIRA, André D. **Segurança da informação no padrão DVB para TV digital**. 2005. 66 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2005-2/adv-tg-final.doc>>. Acesso em: 23 ago. 2006.

WEBREFERENCE. **Introduction to RSS**. [S.l.], 2006. Disponível em: <<http://www.webreference.com/authoring/languages/xml/rss/intro/>>. Acesso em: 31 out. 2006.

WORLD WIDE WEB CONSORTIUM. **Extensible markup language**. [S.l.], 2004.
Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 12 out. 2007.

ZHANG, Yongjun. **A Java 3D framework for digital television set-top box**. 2003. 48 f.
Dissertação (Mestrado em Telecomunicações, Software e Multimídia) – Telecommunications
Software and Multimedia Laboratory, Helsinki University of Technology, Helsinque,
Finlândia. Disponível em: <www.tkk.fi/Units/IDC/brocom/sub/terminal/master/Java_3D_stb.pdf>. Acesso em: 02 nov. 2007.