



**UNIVERSIDADE DO ESTADO DA BAHIA - UNEB
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA – DCET
COLEGIADO DO CURSO DE ANÁLISE DE SISTEMAS**

**POTENCIALIDADE DA TV DIGITAL POR MEIO DE UM
PROTÓTIPO DE APLICATIVO T-COMMERCE**

Tiago Leite Mendes

Orientador: Prof. Dr. Leandro Coelho de Souza

Salvador
Março/2011

TIAGO LEITE MENDES

**POTENCIALIDADE DA TV DIGITAL POR MEIO DE UM
PROTÓTIPO DE APLICATIVO T-COMMERCE**

Trabalho de Conclusão de Curso
apresentado como requisito para
obtenção do grau de Bacharel em
Análise de Sistema.

Orientador: Prof. Leandro Coelho

Salvador
Março/2011

SUMÁRIO

LISTA DE FIGURAS.....	4
LISTA DE SIGLAS.....	6
RESUMO.....	7
ABSTRACT.....	8
1. Introdução.....	9
1.1 TV Digital e Interatividade.....	10
1.2 Componentes da TV Digital.....	11
1.3 Arquitetura da TV Digital.....	15
1.4 Padrões da TV Digital.....	19
1.4.1 Padrão europeu – DVB.....	19
1.4.2 Padrão norte-americano – ATSC.....	20
1.4.3 Padrão japonês – ISDB.....	22
1.5 Padrão Brasileiro e o Middleware Ginga.....	23
2. Middleware Ginga, a camada de software brasileira.....	25
2.1 Arquitetura ginga.....	25
2.1.1 Ginga-CC.....	27
2.1.2 Ginga-NCL.....	27
2.1.3 Ginga-J.....	27
3. Bibliotecas para aplicativos de tv digital.....	30
3.1 Api davic.....	30
3.2 Api havi.....	30
3.3 Api javatv.....	31
3.3.1 Aplicação JavaTV: xlet.....	33
3.4 Emulador xletview.....	36
4 Comércio eletrônico e t-commerce.....	38
5 Trabalhos correlatos.....	41
6 TV-Sellcenter: uma aplicação de t-commerce.....	42
7 Conclusões.....	55
Referências.....	57
Apêndice.....	59

LISTA DE FIGURAS

Figura 1.2.1 – Sistema de TV Digital expandido.....	12
Figura 1.2.2 – Etapas da difusão.....	13
Figura 1.2.3 – Meios de receber sinal.....	14
Figura 1.2.4 – Etapas da recepção.....	14
Figura 1.3.1 – Arquitetura do sistema de TV Digital interativa com tecnologias usadas em cada camada.....	15
Figura 1.3.2 – (a) Sinal original; (b) Sinal distorcido com ruído; (c) Sinal recuperado.....	16
Figura 1.3.3 – Exemplo de aplicação EGP.....	19
Figura 1.4.1.1 – Camadas do padrão DVB-T.....	20
Figura 1.4.2.1 – Camadas do padrão ATSC.....	21
Figura 1.4.3.1 – Camadas do padrão ISDB.....	22
Figura 2.1.1 – Visão geral da arquitetura Ginga.....	26
Figura 2.1.3.1 – Arquitetura Ginga-J e ambiente de execução.....	28
Figura 3.3.1.1 – Escolha de ângulos de câmera.....	34
Figura 3.3.1.2 – Estados de ciclo de vida de um xlet.....	34
Figura 3.3.1.3 – Interface Xlet.....	35
Figura 3.3.1.4 – Interface XletContext.....	36
Figura 3.4.1 – Interface do Xletview.....	37
Figura 6.1 – Arquitetura do funcionamento da camada de aplicação.....	43
Figura 6.2 – Código da implementação da interface Xlet – Modelo Canal de Vendas.....	44
Figura 6.3 – Código da implementação da interface Xlet – Modelo Relacionado Com Programas.....	45
Figura 6.4 – Tela inicial modelo Canal de Vendas.....	47
Figura 6.5 – Tela apresentando informações do produto.....	47
Figura 6.6 – Tela apresentando outros produtos.....	48
Figura 6.7 – Tela apresentando produto número 1.....	49
Figura 6.8 – Tela apresentando cartões para pagamento.....	49

Figura 6.9 – Tela inicial modelo Relacionado com Programas, com a mensagem que indica uma aplicação na programação.....	51
Figura 6.10 – Tela apresentando produto relacionado ao filme.....	51
Figura 6.11 – Tela apresentando informações do produto.....	52
Figura 6.12 – Tela apresentando cartões para pagamento.....	52
Figura 6.13 – Tela apresentando outros produtos relacionados.....	53
Figura 6.14 – Tela apresentando produto 1 – Roupa Matrix.....	53
Figura 6.15 – Tela apresentando produto 2 – Celular Matrix.....	54
Figura 8.1 – Diagrama de Classes – ShoTv.....	59
Figura 8.2 – Diagrama de Classes – Matrix.....	61

LISTA DE SIGLAS

8-VSB	– 8 Level – Vestigial Side Band Modulation
API	– Applications Programming Interface
ARIB	– Association of Radio Industries and Businesses
ATSC	– Advanced Television Systems Committee
COFDM	– Coded Orthogonal Frequency Division Multiplexing
DASE	– Digital Applications Software Environment
DAVIC	– Digital Audio Visual Council
DVB	– Digital Video Broadcasting
EGP	– Electronic Program Guide
HAVi	– Home Audio Video Interoperability
ISDB	– Integrated Services Digital Broadcasting
JMF	– Java Media Framework
MPEG	– Picture Experts Group
MHP	– Multimedia Home Platform
NCL	– Nested Context Language
SBTVD	– Sistema Brasileiro de TV Digital

RESUMO

Um novo modelo de transmissão de sinal de televisão foi implantado no Brasil, a transmissão de sinal digital, ou seja, a TV Digital. Essa tecnologia de TV proporciona melhor qualidade de imagem e som e dá suporte a transmissão de dados, possibilitando a interatividade do telespectador com a programação da emissora. Vários tipos de aplicações interativas poderão ser desenvolvidas, como exemplo aplicações de enquetes, acesso à internet, aplicações de comércio eletrônico e outros. Este trabalho tem como finalidade explorar a potencialidade deste novo meio de mídia através de um protótipo de aplicativo *t-commerce*, onde *t-commerce* é um comércio eletrônico efetuado pela TV. O protótipo desenvolvido destaca a ideia de como o telespectador interage em uma aplicação de comércio eletrônico, no qual ele visualiza informações do produto apresentado e, se desejar, realiza a compra, tudo por meio do controle remoto.

Palavras – chave: TV Digital, API JavaTV, T-Commerce, Xletview.

ABSTRACT

A new model of television signal transmission was implemented in Brazil, the digital signal transmission, or Digital TV. This technology provides better quality television picture and sound, and supports data transmission, allowing for viewer interactivity with the programming of the station. Various types of interactive applications may be developed, such as applications of polls, internet access, e-commerce applications and more. The paper aims to explore the potential of this new medium of media through an application prototype t-commerce, where t-commerce is commerce conducted by a TV. The prototype highlights the idea of how the viewers interacts with an application of electronic commerce in which to view product information submitted and, if desired, makes the purchase, all by remote control.

Key-Words: Digital TV, API JavaTV, T-Commerce, Xletview.

1. Introdução

Um dos meios de comunicação mais difundidos e importantes do mundo, a televisão, evolui tecnologicamente para melhor. Depois de passar da transmissão em preto e branco para a transmissão colorida, a televisão evolui da tecnologia analógica para a tecnologia digital, a TV Digital.

Esse novo sistema de TV traz consigo uma melhora significativa na qualidade de imagem e som transmitidos, ou seja, ausência de ruídos, chuviscos, que ocorre na transmissão analógica. No entanto, com a TV Digital é possível, também, a transmissão de dados. A TV Digital proporciona a interatividade, no caso, a possibilidade do telespectador interagir com a TV.

O sinal digital no Brasil começou a ser transmitido em dezembro de 2007, sendo que após anos de estudos sobre os padrões de TV Digital que adotaria, o Brasil decidiu criar seu próprio padrão de TV Digital, com *middleware* Ginga, onde Ginga é o nome dado à camada de software presente no receptor de TV Digital, responsável pela execução das aplicações interativas; e foi desenvolvido para atender as necessidades de inclusão social/digital. (BRACKMANN, 2008)

Com algumas ferramentas, como a JavaTV, é possível desenvolver aplicações para TV Digital que contemplem o que esta se propõe, a interatividade. Entre as aplicações que serão possíveis está o comércio eletrônico pela TV, denominado *T-commerce*.

Além de descrever as características envolvidas no sistema de TV Digital, o objetivo deste projeto é explorar a potencialidade da TV Digital por meio de um protótipo de aplicação *t-commerce*. Este aplicativo é desenvolvido com o uso da API JavaTV e outras, sendo simulado no emulador Xletview. O aplicativo é apresentado em duas partes, demonstrando dois tipos de *t-commerce* que serão possíveis. O primeiro tipo é o *t-commerce* sendo usado em um canal de vendas, o segundo é o comércio sendo feito no momento que telespectador assiste a um filme.

Este documento começa por explicar o que é TV Digital e interatividade, mostrando seus componentes e sua arquitetura, abordando, também, os padrões existentes e o padrão brasileiro de TVD (capítulo 1). Em

outras seções aborda detalhes sobre o *middleware* Ginga (capítulo 2), sobre algumas bibliotecas para desenvolvimento de aplicações (capítulo 3), faz uma abordagem sobre comércio eletrônico e o *t-commerce* (capítulo 4) e finaliza por mostrar a aplicação desenvolvida neste projeto (capítulo 6).

1.1. TV Digital e Interatividade

Mais do que qualidade de imagem e som, a TV Digital traz consigo a possibilidade de interação. O telespectador deixa de ser um usuário passivo, passa agora a ter a possibilidade de interagir com o conteúdo que está sendo transmitido. Por meio de uma interface especial, como, por exemplo, o controle remoto, o telespectador poderá acessar a Internet, verificar e-mails, conferir saldo bancários, fazer compras, responder enquetes e muito mais.

Contudo, a TV Digital apresenta outros benefícios que vão muito além da interatividade, como por exemplo: alterações no padrão de exibição dos programas poderão ser exibidos em *widescreen*, ou seja, formato de tela 16:9, podendo chegar a 1920x1080 pixels, significando imagens de alta definição; a qualidade de transmissão com portabilidade e mobilidade, i.e., qualquer dispositivo portátil, como celulares, notebook, *smartphones*, receberiam transmissão digital com a possibilidade de assistir TV em qualquer lugar, mesmo estando em movimento, como em um carro, com som e imagem excelentes; a multiprogramação, em que a emissora transmitirá vários programas ao mesmo tempo ou vários ângulos de um mesmo programa, ficando à disposição do telespectador escolher a melhor opção. Com todas as novidades trazidas pela TV Digital, a interatividade é, sem dúvida, a mais atrativa e vem sendo estudada/discutida pela comunidade acadêmica e/ou comercial podendo ser classificada em três categorias, de acordo com CANDIDO JUNIOR (2008):

- **Local:** Junto com a transmissão normal de um canal televisivo, é adicionado conteúdo do tipo imagens de baixa definição, textos ou aplicativos enviados pelas emissoras. Não há necessidade de canal de retorno.

- **Intermitente:** Nesta classificação há necessidade de um canal de comunicação entre o telespectador e a emissora. Este canal de retorno (e.g., linha telefônica), é necessário para garantir a interatividade do telespectador que deverá utilizá-lo para manifestar suas ações e acaba gerando gastos adicionais ao telespectador. É intermitente, pois não funciona o tempo todo, apenas quando o usuário se manifesta através de comandos interativos.
- **Permanente:** Esta última classificação garante ao telespectador um canal de retorno dedicado 100% do tempo, permitindo assim uma interatividade em tempo real. Nesta configuração é permitido o acesso à Internet, mensagens instantâneas, serviços bancários, serviços de governo, comércio eletrônico, jogos e transmissão de vídeos por *streaming*.

Não obstante, o canal de retorno ou canal de interatividade é o meio que faz a conexão entre a emissora e o telespectador. É o canal que possibilita à emissora receber dados que foram enviados dos aparelhos de TVs e conversores digitais. (GUILHERME e GOMARIN, 2009) Como tecnologias, o canal de retorno pode usar telefonia móvel, ADSL, satélite, plataformas SMS, canal de frequência intrabanda ou outras formas disponíveis. (SANTOS, 2007).

A estrutura física e o funcionamento da TV Digital serão mostrados com mais detalhes na próxima seção.

1.2. Componentes da TV Digital

De acordo com MONTEZ e BECKER (2005), um sistema de TV Digital interativa está dividido basicamente em três componentes:

- **Difusor** – parte responsável por prover conteúdo a ser transmitido aos receptores. O difusor provê dois tipos de serviços: Provedor de Serviços de Difusão, que trabalha com o conteúdo que será transmitido; e o Provedor de Serviços de Interação, que trabalha com as entradas do canal de interação.

- **Receptor** – componente responsável por receber o conteúdo transmitido pelo difusor e por meio deste o telespectador pode reagir ou interagir com o difusor.
- **Meio de difusão** – é o componente que faz a conexão entre o difusor e o receptor. Entre os meios de difusão mais comuns estão: satélites, cabo dedicado e radiodifusão.

A figura 1.2.1 abaixo mostra um esquema contendo os três componentes básicos de um sistema de TV Digital.

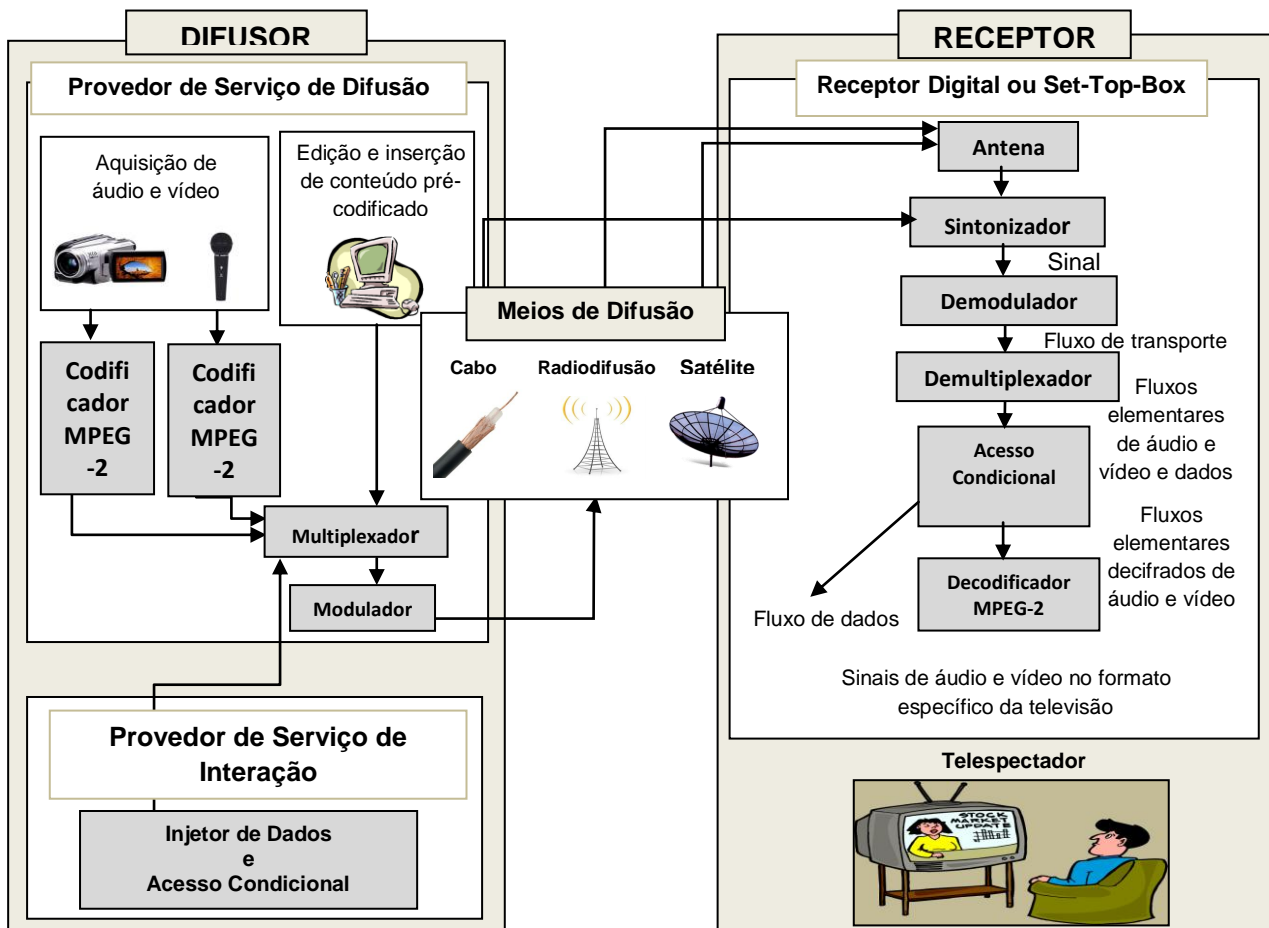


Figura 1.2.1: Sistema de TV Digital expandido (MONTEZ e BECKER, 2005)

A figura descreve as etapas que ocorrem no processo de difusão e no processo de recepção das informações. Pode-se notar que no difusor os sinais de vídeo e áudio e dados são codificados no padrão MPEG-2 (*Picture Experts Group*) e encapsulados em pacotes de transporte MPEG2-TS, gerando fluxos elementares. Esses fluxos são multiplexados, gerando apenas um fluxo contendo todos os serviços, que também é codificado e transportado. Após

isso, passa pelo processo de modulação, que permite a difusão pelos meios convencionais (Figura 1.2.2). Neste caso já pode ser vista a diferença entre o sinal analógico e o sinal digital. No caso de transmissão digital, a quantidade de conteúdo enviado é maior do que no analógico, visto que a transmissão digital pode ser compactada possibilitando que mais conteúdos utilizem o mesmo canal de transmissão. Na questão da transmissão dos dados acontece um processo no difusor que é chamado por MONTEZ e BECKER (2005) de carrossel de dados. Neste processo a aplicação é dividida em partes ou módulos que são enviados de forma periódica. No caso da interação os módulos são enviados e guardados na memória do receptor até que todo o arquivo esteja completo para ser executado. O receptor pode estar dentro do aparelho de televisão, no caso dos aparelhos mais modernos, ou pode ser um aparelho a parte que faz a ponte entre a televisão e a fonte externa de sinal. Neste caso o receptor seria conhecido como set-top-box. O set-top-box tem como função básica converter sinais digitais para que possam ser assistidos em aparelhos analógicos (MONTEZ e BECKER, 2005).

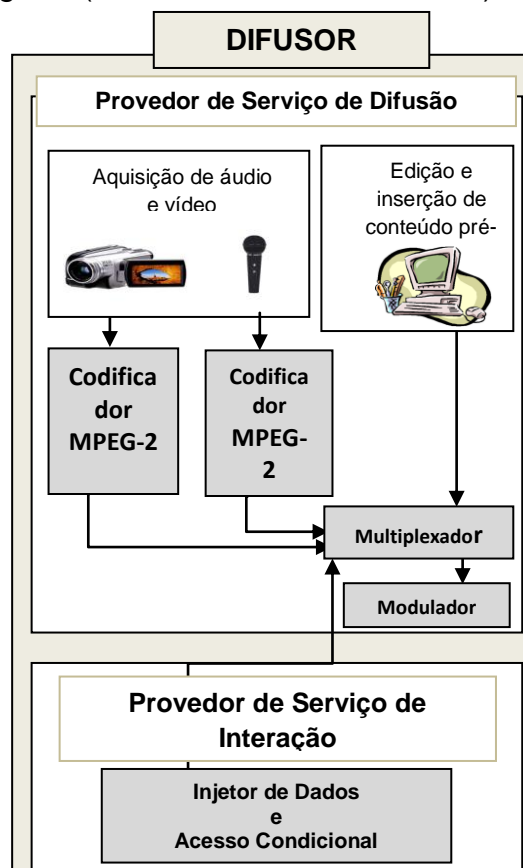


Figura 1.2.2: Etapas da difusão. (MONTEZ e BECKER, 2005)

O pacote enviado é então recebido por uma fonte externa de sinal, que pode ser uma antena específica ou via cabo (Figura 1.2.3), e passado para o receptor que faz o processo inverso do difusor, demodular – extrair o fluxo de transporte MPEG-2 -, demultiplexar – extrair os fluxos elementares - e decodificar – converter para o sinal apropriado de exibição - o sinal para todos os fluxos, levando a informação até o telespectador (Figura 1.2.4).



Figura 1.2.3: Meios de receber sinal. (MONTEZ e BECKER, 2005)

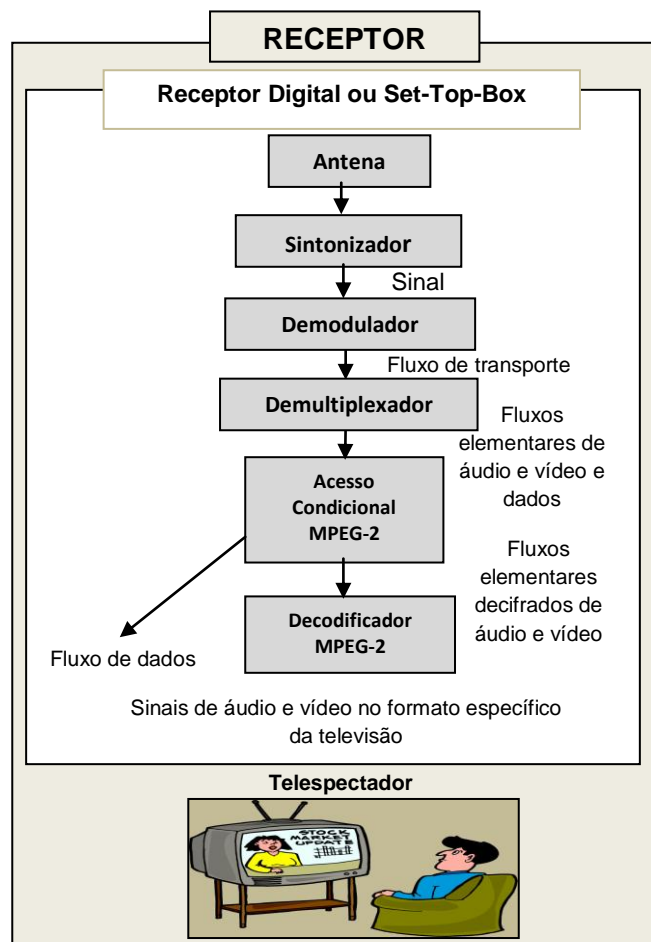


Figura 1.2.4: Etapas da recepção (MONTEZ e BECKER, 2005)

Por se encarregar de uma tarefa complexa, demodular, demultiplexar sinal e processar dados, o set-top-box possui processador, memória, sistema operacional, uma camada de software chamada *middleware*, entre outras. Dentre esses o *middleware* é de suma importância, pois é responsável por dar suporte à interatividade, incluindo a execução das aplicações.

No caso da interatividade, os dados são enviados para o provedor de serviços por meio do set-top-box, que usa um canal de retorno, que pode ser telefonia fixa, telefonia móvel, tecnologia ADSL, rádio e outros.

Mais detalhes sobre o *middleware* e os outros componentes da arquitetura do sistema de TV digital serão mostrados na seção posterior.

1.3. Arquitetura da TV Digital

A arquitetura do sistema de TV Digital é semelhante ao sistema de redes, ou seja, baseada em camadas, onde cada camada dá suporte à camada superior e utilizam serviços da anterior, sem se preocupar com a forma com que a camada inferior tratou os dados. A figura 1.3.1 mostra a arquitetura de um sistema digital interativo de forma geral.

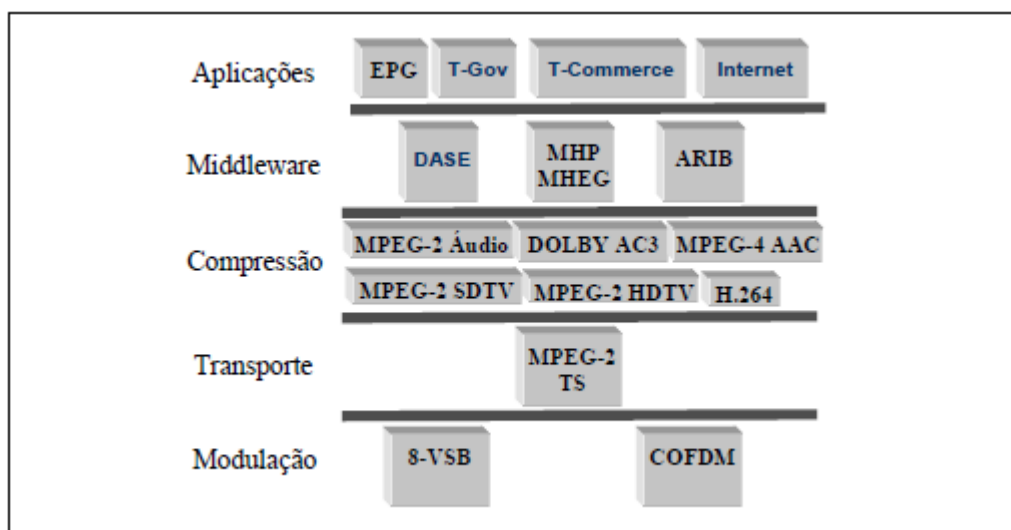


Figura 1.3.1: Arquitetura do sistema de TV digital interativa com tecnologias usadas em cada camada. (MONTEZ e BECKER, 2005)

Devido aos problemas que ocorrem nos canais de comunicação, cabos, ondas de rádio e outros, causando ruídos, interferências e outras

perdas no sinal, é necessária a modulação do sinal. A camada de **Modulação** é responsável por adaptar o sinal para que sofra o mínimo de distúrbios, interferências ou outro problema ao ser transmitido.

De acordo com MONTEZ e BECKER (2005) a modulação oferece três benefícios: o conteúdo espectral de um sinal de mensagens é deslocado para dentro da faixa de frequência de um canal de comunicação; o conteúdo do sinal será colocado numa forma que sofrerá menos interferências durante a transmissão; e permite compartilhar os enlaces de comunicação transmitindo dados simultaneamente.

Existem três formas ou técnicas de modulação de sinal: por frequência, por amplitude e por fase. Na modulação por frequência (FM) a frequência da portadora (sinal a ser modulado) é alterada de acordo com a informação a ser transmitida. Na modulação por amplitude (AM) a amplitude da portadora varia em função da modulação do sinal. E na modulação por fase (PM) a fase da portadora é variada de acordo com os dados a serem transmitidos. (PROTEVE, 2010)

Na transmissão digital esses métodos normalmente são combinados para obter transmissão menos propensa a erros e para aumentar a capacidades de transmissão de bits por segundo (MONTEZ e BECKER, 2005). Por exemplo, na AM duas amplitudes diferentes são usadas para representar 0 e 1 respectivamente, e na PM a portadora pode ser deslocada de 45, 135, 225 ou 315 graus para transmitir dois bits de informações por intervalo de tempo (TANENBAUM, 2003). Na figura 1.3.2, pode-se ver a facilidade de recuperação de um sinal digital distorcido com ruído.

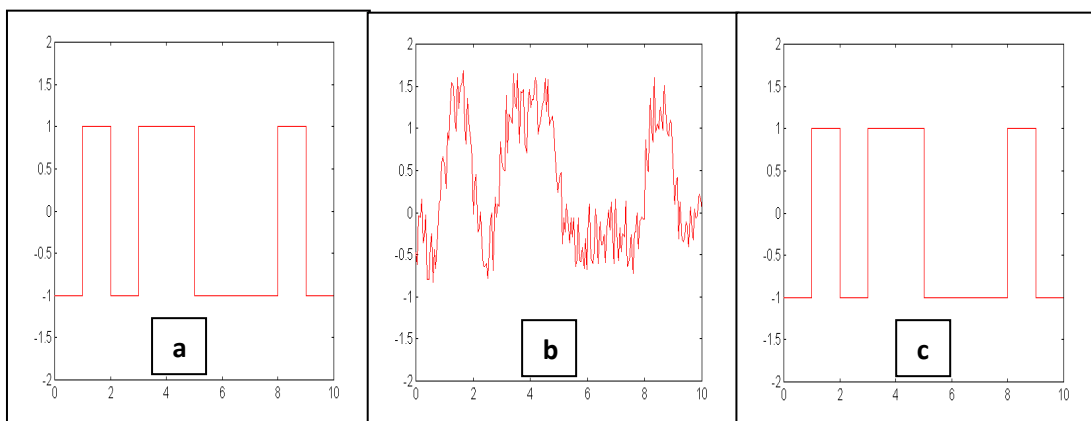


Figura 1.3.2: (a) Sinal original; (b) Sinal distorcido com ruído; (c) Sinal recuperado.

Os padrões de modulação para o sistema de TV Digital usados atualmente são: 8-VSB (*8 Level – Vestigial Side Band Modulation*), que faz a modulação por amplitude, onde utiliza 8 níveis discretos de ondas de radiofrequência para conduzir seus dados; e o COFDM (*Coded Orthogonal Frequency Division Multiplexing*), que faz a combinação de modulação por amplitude e por fase, com múltiplas ondas portadoras transportando dados. (MONTEZ e BECKER, 2005).

Na camada de **Transporte** se encontra o padrão MPEG-2 TS (*Moving Picture Experts Group*) (*Transport Stream*) que é responsável pelo transporte ou difusão de arquivos multimídia. O tamanho do pacote é de 188 bytes, que serve para facilitar a resincronização de um arquivo, por exemplo, vídeo, no caso de perda de pacotes.

Na camada de **Compressão** estão os padrões de compressão de áudio e vídeo. Entre os de áudio encontram-se MPEG-2 Audio, DOLBY AC3, MPEG-4 ACC. Os padrões de vídeo presentes estão o MPEG-2 SDTV, MPEG-2 HDTV e H.246, conforme visto na figura 1.3.1.

A compressão de dados é importantíssima na questão do processamento de mídias digitais. Na TV Digital é essencial que arquivos de áudio e vídeos passem por esse processo para que sejam difundidas. Sabe-se que arquivos de vídeo e áudio digitais não comprimidos possuem elevados requisitos de armazenamento e largura de banda, tornando inviável a sua transmissão.

A compressão é feita por meio de duas técnicas que exploram as seguintes características: redundância e percepção humana. A técnica de compressão por redundância baseia-se na eliminação de informações redundantes, por meio de técnicas de compressão, para diminuir o tamanho do arquivo original. Por exemplo, em um arquivo de vídeo, um cenário fixo apresenta uma serie de redundâncias que podem ser exploradas; em um arquivo de áudio a redundância é normalmente explorada nas partes de silêncio. A outra técnica usada é a percepção humana que explora os estímulos do sistema nervoso para comprimir os arquivos. Como exemplos, algumas frequências de áudio não são perceptíveis a audição humana; o olho humano é mais sensível a variações de tons de cinza do que a variações de

tons coloridas; para comprimir os dados aplicam-se vários métodos que exploram essas características fazendo, assim, diminuir o tamanho do arquivo original (MONTEZ e BECKER, 2005).

A camada que faz a conexão do hardware e o sistema operacional do set-top-box com as aplicações é a camada **Middleware**. É uma camada de software que é responsável pela execução das aplicações, controle e monitoramento dos recursos do terminal e disponibilizá-los às aplicações. Qualquer aplicação pode ser executada nesta camada independente do fabricante do set-top-box, desde que este seja compatível com o padrão de *middleware*. Nesta camada está presente APIs que controlam as aplicações. (GATIS, 2006).

Entre os padrões de *middleware* atualmente difundidos estão o MHP (*Multimedia Home Platform*), que faz parte do padrão europeu de TV Digital; o DASE (*Digital Applications Software Environment*), *middleware* do padrão norte-americano; e o ARIB (*Association of Radio Industries and Businesses*), *middleware* do padrão japonês. Mais detalhes a respeito destes *middlewares* serão abordados na seção seguinte.

Finalmente, a camada de **Aplicação** é a camada que proporciona os serviços da TV Digital interativa. Dentre esses serviços estão internet, *t-banking*, *t-commerce*, *t-gov*, guias de programação eletrônica (EGP) e outros. Todas essas aplicações são carregadas no set-top-box, no caso das aplicações que permitam interação com o telespectador, será usado o canal de retorno do set-top-box. A empresa de TV por assinatura do Brasil via satélite, SKY, apresenta um exemplo de uma aplicação EGP, como pode ser vista na figura 1.3.3.



Figura 1.3.3: Exemplo de aplicação EGP (TV Magazine, 2009)

1.4. Padrões da TV Digital.

Apesar de se ter iniciado as pesquisas em desenvolvimento da TV digital no fim da década de 1980 apenas em 1993 foram lançadas os primeiros padrões de transmissão de TV Digital: o DVB, sistema europeu e o ATSC, padrão norte-americano. Apenas em 1999 o Japão lançou o padrão ISDB.

1.4.1. Padrão europeu – DVB

DVB ou *Digital Video Broadcasting* é o padrão europeu de televisão digital. Tem como característica o foco na interatividade junto com a qualidade de imagem de alta definição, possui também suporte para transmissão em dispositivos móveis, porém, em uma banda extra, o DVB-H.

Como tecnologias de transmissão o DVB desenvolveu três padrões: DVB-T, radio-difusão terrestre; DVB-S, difusão via satélite e DVB-C, difusão via cabo. Para a modulação utiliza o padrão COFDM, explicado anteriormente, com taxa de transmissão entre 5 a 31,7 Mbits. Arquivos de vídeo e áudio são multiplexados e codificados no padrão MPEG-2.

O *middleware* do DVB é o MHP que foi desenvolvido em Java e usa modelos de aplicação definidos pelo JavaTV. As aplicações suportadas podem ser desenvolvidas tanto em Java como em HTML, isso é possível porque o MHP possui APIs Java que dão todo o suporte para várias funcionalidades. (BORDERS, 2010). O DBV-T também usa o padrão MHEG, que é usado para representar conteúdos multimídia e permite a interatividade do telespectador com a aplicação. (MONTEZ e BECKER, 2005). Sua arquitetura é mostrada na figura 1.4.1.1 abaixo.

Segundo análise de pesquisadores, caso o Brasil adotasse esse sistema as empresas de telefonia celular poderiam investir em aparelhos celulares para oferecer serviços de TV Digital, no entanto, de acordo com o Site Oficial da TV Digital Brasileira (DTV, 2010), a transmissão para dispositivos móveis é tarifada pelas operadoras de telefonia no padrão europeu, já o sistema brasileiro pretende oferecer este serviço sem custos ao consumidor.

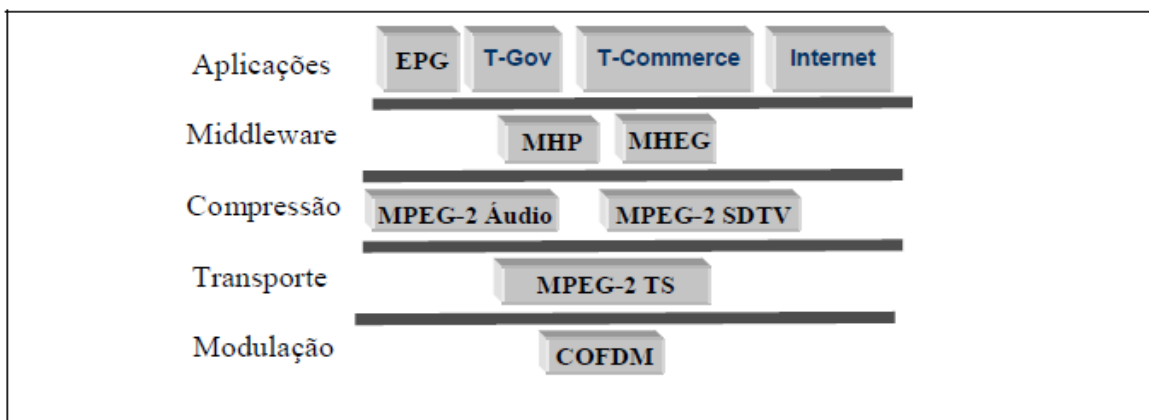


Figura 1.4.1.1: Camadas do padrão DVB-T. (MONTEZ e BECKER, 2005)

1.4.2. Padrão norte-americano – ATSC

O padrão norte-americano de TV Digital *Advanced Television Systems Committee* (ATSC) apesar de dar suporte à transmissão de imagem de alta definição, não oferece transmissão digital a dispositivos móveis.

Os padrões de transmissão são o ATSC-C, transmissão via cabo e o ATSC-T, via rádio-difusão. Para a modulação de sinal utiliza o padrão 8-VSB,

com transmissão de 19,39 Mbps com canal de 6 MHz. O padrão MPEG-2 é utilizado para multiplexação e codificação de vídeo e para o áudio utiliza o padrão Dolby AC-3. (ATSC, 2007)

O *middleware* desenvolvido e utilizado pelo ATSC é o DASE. É semelhante ao MHP no sentido de suportar aplicações procedurais desenvolvidas em Java, pois dá suporte a Máquina Virtual Java, e também suporta aplicações declarativas, desenvolvidas em HTML e JavaScript. Sua arquitetura é vista abaixo, na figura 1.4.2.1.

Algumas vantagens deste sistema seriam o menor custo de implantação, oferecendo maior acessibilidade à população, pois os custos dos aparelhos com conversores deste sistema de TV Digital são menores. Como também, a possibilidade de exportação de aparelhos compatíveis fabricados localmente, visto que quase todos os aparelhos comercializados nos Estados Unidos são importados, neste caso as empresas fabricantes poderiam investir na adoção deste padrão no Brasil.

No entanto, durante o processo de escolha do sistema brasileiro, os estudos mostraram que o sistema norte-americano é o menos adequado para atender as necessidades nacionais, tendo em vista que um dos fatores determinantes na escolha do sistema nacional foi a possibilidade de transmissão em dispositivos móveis e portáteis.

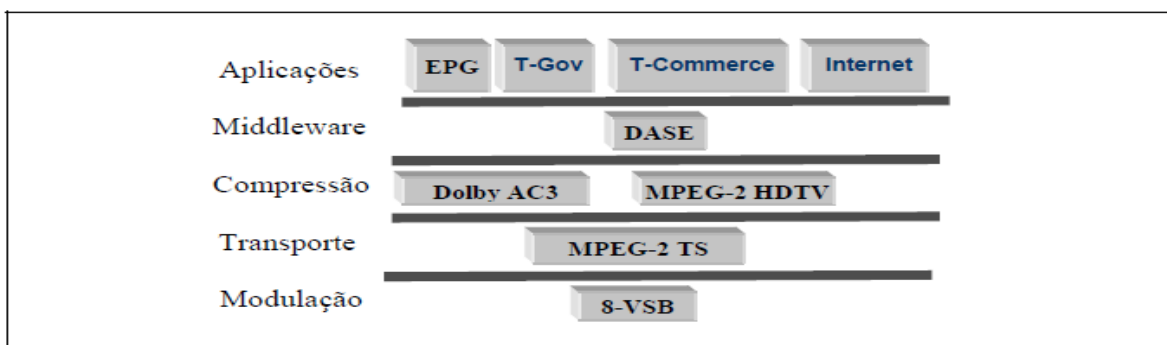


Figura 1.4.2.1: Camadas do padrão ATSC. (MONTEZ e BECKER, 2005)

1.4.3. Padrão japonês – ISDB

O padrão de transmissão de TV Digital japonesa o *Integrated Services Digital Broadcasting* (ISDB) tem como vantagem principal suporte para transmissão digital para dispositivos móveis e portáteis dentro da faixa que é usada para transmissão terrestre, desde que use broadcast.

Este padrão usa três tipos de tecnologias para transmissão: ISDB-T, transmissão terrestre; ISDB-S, via satélite e ISDB-C, transmissão via cabo.

O padrão de modulação usado é o mesmo do padrão europeu, o COFDM, possuindo certas variações, utilizando uma taxa de transferência entre 3.65 e 23.23 Mbps com canal que varia entre 6 a 8 MHz. O padrão MPEG-2 é utilizado para codificar e multiplexar arquivos de vídeo, para arquivos de áudio usa o MPEG-2 AAC. (DIBEG, 2009)

O *middleware* escolhido pelo padrão japonês é o ARIB. Este é composto de especificações que dão suporte a aplicações baseadas em Java, e também dá suporte a linguagens declarativas como a BML (*Better Markup Language*) que é baseada em XML (*Extensible Markup Language*). (ARIB, 2004) A figura 1.4.3.1 mostra a arquitetura ISDB.

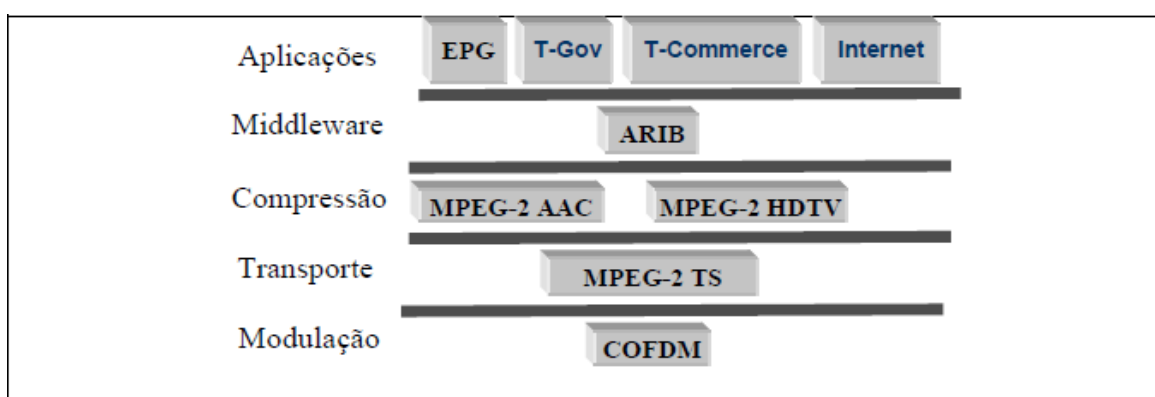


Figura 1.4.3.1: Camadas do padrão ISDB. (MONTEZ e BECKER, 2005)

Estes três padrões são os principais em funcionamento, sendo o último o mais eficiente e completo (GUILHERME e GOMARIN, 2009). Na escolha do sistema brasileiro, concluiu-se que o sistema que melhor se adequava era o ISDB, pois, um dos fatores importantes na escolha foi a possibilidade de transmissão para dispositivos móveis e portáteis, sendo assim, o padrão

japonês serviu de base para o SBTVD (Sistema Brasileiro de TV Digital), conhecido também por ISDB-TB (*Integrated Services Digital Broadcasting – Terrestrial*) (DTV, 2010). O padrão brasileiro desenvolveu seu próprio *middleware*, o GINGA. Este *middleware* oferece todas as facilidades para aplicativos de interação para a TV Digital.

1.5. Padrão brasileiro e o middleware Ginga

Em 26 de novembro de 2003 foi assinado um Decreto, nº 4.901, pelo presidente Luis Inácio Lula da Silva que criou o sistema brasileiro de televisão digital. Em dezembro de 2007 começou a transmissão do sinal de TV Digital no Brasil. A previsão é de que em junho de 2016 todo o país esteja recebendo o sinal digital em seus televisores e o sinal analógico seja extinto.

Desenvolvido com base no padrão japonês, o ISDB-TB ou Sistema Brasileiro de TV Digital introduziu tecnologias modernas para a compressão de vídeo e áudio, tornando-se superior ao padrão japonês neste sentido.

De acordo com o Site Oficial da TV Digital Brasileira (2010), o padrão brasileiro dá suporte à interatividade, além de transmissão de alta qualidade em imagem e som com suporte à mobilidade e portabilidade.

O que realmente mudou foram os padrões de modulação e codificação. No caso da codificação de vídeo, o padrão usado é superior aos demais usados nos sistemas de TV Digital internacionais. Assim como os outros sistemas este também é baseado em camadas, cada camada tendo seu padrão para modulação, compressão e transporte e *middleware*. A única similaridade com os outros padrões de transmissão é na camada de transporte onde o SBTVD usa o padrão MPEG-2 TS. Já nas outras camadas usa tecnologias diferentes, como por exemplo, para a modulação utiliza o padrão BST-OFDM (*Band Segmented Transmission - Orthogonal Frequency Division Multiplexing*) que divide o canal em segmentos onde cada segmento contém parte da informação. Esses segmentos são transmitidos de forma paralela, para que o mínimo de informação seja perdido na transmissão. No entanto “*um segmento OFDM deve obrigatoriamente ter uma configuração que permita a conexão de múltiplos segmentos para prover uma largura de transmissão que*

atenda à necessidade da mídia.” (ABNT NBR, 2007). A segmentação da banda utilizada é feita pelo BST, que pode dividir um espectro de 6 MHz em três partes. (BRACKMANN, 2008).

Na codificação e compressão de arquivos de áudio e vídeo faz-se uso de tecnologias que possibilitam imagem e som de alta qualidade em aparelhos portáteis. Para a codificação de vídeo o SBTVD adotou o padrão H.264, novo padrão de qualidade de vídeos de alta definição que permite visualização da imagem tanto em telas de alta resolução quanto em aparelhos portáteis. Os perfis utilizados são *high profile*, com nível 4.0, ou seja, bit rate máximo de 25 Mbit/s; e o *baseline profile*, que oferece nível máximo de 1.3, bit rate máximo de 768 kbit/s (BRACKMANN, 2008). Os níveis indicam os tipos de receptores. O nível 4.0 é usado em transmissões de TV de alta resolução, além de discos Blu-Ray e HD-DVD; o nível 1.3 é utilizado em receptores que com baixo recurso computacional, por exemplo, dispositivos móveis.

O padrão usado para codificação de áudio é o MPEG-4 AAC (BRACKMANN, 2008), com perfis permitidos: LC (*Low Complexity*), perfil básico do AAC, com níveis L2 e L4, dois canais e multicanal, respectivamente; HE (*High Efficiency*), perfil de alta eficiência, na versão 1 combina o LC com a ferramenta SBR (*Spectral Band Replication*), também utiliza dois canais L2 e L4; e na versão 2, HE combinado com a ferramenta PS (*Parametric Stereo*), com canal nível L2. (ABNT NBR, 2007). O padrão se compromete a codificação de áudio resultando em alta qualidade tanto em dispositivos fixos quanto portáteis.

Na camada de software o Brasil desenvolveu seu próprio middleware: o Ginga, o qual oferece suporte às aplicações interativas independentes dos fabricantes do hardware. De acordo com o site oficial Ginga (2010), o *middleware “é constituído por um conjunto de tecnologias padronizadas e inovações brasileiras que o tornam a especificação de middleware mais avançada do mundo atualmente e a melhor solução para os requisitos do país.”* Uma explanação mais profunda a respeito do Ginga será discutida no próximo capítulo.

2. Middleware Ginga, a camada de software brasileira

Desenvolvido inteiramente no Brasil pelo trabalho em conjunto da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) e Universidade Federal da Paraíba (UFPB), o *middleware* Ginga, que é um software livre, leva em conta as necessidades do país, ou seja, inclusão social e digital se propondo a levar aos cidadãos, por meio do canal mais difundido do país, a TV, informação, educação e serviços sociais. Entre esses serviços seriam oferecidos serviços de saúde, serviços do governo, serviços de EAD, serviços bancários, na TV Digital esses serviços interativos seriam conhecidos como *T-Health*, *T-Government*, *T-Learning* e *T-Banking*.

O middleware brasileiro já está sendo implantado nos conversores de TV Digital. Atualmente foi posta à venda televisores com Ginga embutido nos conversores. De acordo com o site IDG Now (2010), a LG iniciou a venda de televisores com o Ginga para a TV Digital interativa.

2.1. Arquitetura Ginga

O middleware Ginga se subdivide em três subsistemas principais interligados: Ginga-NCL, Ginga-J e Ginga-CC, sendo os dois primeiros os principais na questão da infra-estrutura para a programação das aplicações. A escolha desses paradigmas depende das funcionalidades requeridas em cada aplicação. Um pode ser melhor do que o outro dependendo dessas funcionalidades. A figura 2.1.1 apresenta uma visão geral da arquitetura Ginga.

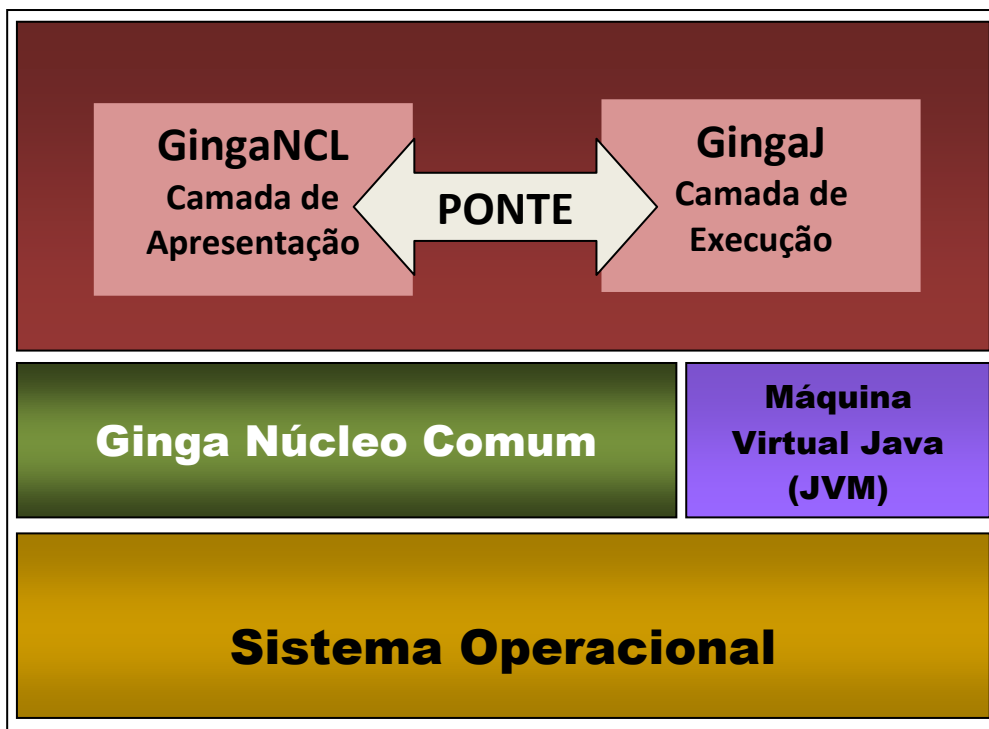


Figura 2.1.1: Visão geral da arquitetura GINGA. (OPENGINGA, 2009)

As duas estruturas principais representam estruturas de programação do tipo declarativa (Ginga-NCL) e procedural (Ginga-J), estas não são totalmente independentes, a arquitetura GINGA permite uma API de ponte que permite que aplicações procedurais utilizem serviços providos por aplicações declarativas ou vive-versa.

Uma linguagem procedural é uma linguagem baseada em módulos, orientada a objetos, onde o programador tem todo o controle sobre seu código – fluxo de controle e execução de seu programa.

Uma linguagem declarativa “é uma linguagem que enfatiza a descrição declarativa do problema”. (ABNT NBR, 2010). Baseada em documento hipermídia, composto por documentos declarativos. Os elementos que compõem a aplicação são descritos por esses documentos e são escritos por uma linguagem de marcação do tipo XML.

2.1.1. Ginga-CC

Núcleo Comum Ginga (*Ginga Common Core*), tem como finalidade obter conteúdos transportados em fluxos de transporte MPEG-2 TS e, por meio de decodificadores de conteúdos comuns, tratar e exibir objetos de mídia, como JPEG, MPEG-4, GIF, PNG e outros. Também controla o acesso ao canal de retorno e ao sistema operacional. O Ginga-CC dá suporte aos ambientes Ginga-NCL e Ginga-J.

2.1.2. Ginga-NCL

Para aplicações declarativas foi desenvolvido pela PUC-Rio o subsistema Ginga-NCL, que utiliza linguagem declarativa.

É responsável pelo processamento de documentos hipermídia escritos em NCL (*Nested Context Language*). A linguagem NCL “*é uma linguagem de aplicação XML com facilidades para a especificação de aspectos de interatividade, sincronismo espaço-temporal entre objetos de mídia, adaptabilidade, suporte a múltiplos dispositivos e suporte à produção ao vivo de programas interativos não-lineares.*” (GINGA-NCL, 2008).

Os componentes principais do subsistema Ginga-NCL são uma máquina de interpretação de conteúdos declarativos NCL, exibidor XHTML, que inclui interpretadores CSS e ECMAScript e a máquina de apresentação Lua, responsável pela interpretação dos scripts Lua.

2.1.3. Ginga-J

Para as aplicações procedurais foi desenvolvido pela UFPB o subsistema Ginga-J que é responsável pelo processamento de conteúdos ativos. Executa aplicações desenvolvidas em Java, voltadas especificamente para ambiente de TV Digital.

A arquitetura Ginga-J se distingue em entidades ou recursos de hardware, sistema do software e aplicativos, conforme mostrado na figura 2.1.3.1 abaixo.

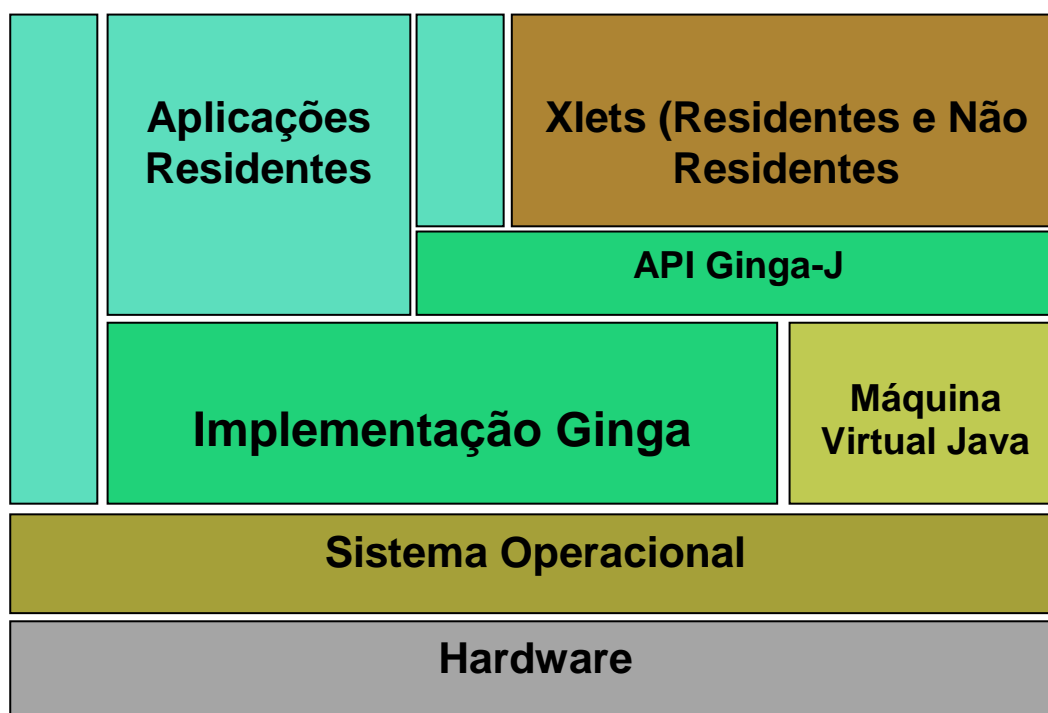


Figura 2.1.3.1: Arquitetura Ginga-J e ambiente de execução (SOUZA FILHO, LEITE e BATISTA, 2007).

De acordo com a SOUZA FILHO, LEITE e BATISTA (2007) as aplicações residentes (*Native Applications*) podem ser implementadas utilizando funções não padronizadas, que o sistema operacional do dispositivo de Ginga fornece, ou por implementação particular do Ginga. A API padronizada Ginga fornece funcionalidades que podem ser absorvidas pelos aplicativos residentes. E os aplicativos transmitidos (Xlets) sempre devem utilizar API padronizada fornecida pelo Ginga-J.

As aplicações residentes têm prioridade sobre aplicações Ginga, entre elas estão: *closed caption*, mensagens do sistema de acesso condicional, guias de programação eletrônica residente, entre outras.

A especificação Ginga – J é uma adaptação da API de acesso a informação de serviço do padrão japonês ISDB ARIB B.23 com a especificação da API Java DTV, que inclui a API JavaTV (OPENGINA, 2009). Faz também

uso de APIs para o processamento de classes compiladas, entre elas estão, API Davic, API HAVi e API DBV. As características dessas APIs serão explicadas mais tarde. O componente principal neste subsistema é o mecanismo de execução do conteúdo procedural baseado em máquina virtual Java.

3. Bibliotecas para aplicativos de TV Digital

Neste capítulo serão apresentadas ferramentas utilizadas para o desenvolvimento de aplicativos para a TV Digital, ou seja, um conjunto de APIs e seus conceitos, entre elas, estão API Davic, API HAVi, API JavaTV. Essas APIs foram desenvolvidas para trabalharem em conjunto na construção desses tipos de aplicações, pois elas dão suporte à apresentação gráfica e tratamento de eventos, entre outros. É apresentado também o conceito de Xlet e o emulador que será utilizado para os testes do aplicativo *t-commerce* que será mostrado neste trabalho, o emulador Xletview.

3.1. API Davic

A associação DAVIC (*Digital Audio Visual Council*) desenvolveu este conjunto de especificações “*com o objetivo de promover o sucesso de aplicações interativas digitais áudio-visuais*” (DAVIC, 2009) e interoperabilidade para plataformas que executam vídeo e áudio transmitidos via broadcast. A ideia é divulgar conteúdo multimídia interativo ao maior público possível, independente de fronteira geográfica, aplicativos, serviços e outros.

As especificações Davic foram até a versão 1.4, e terminou em 1999. A versão 1.5 adicionou ferramentas e definiu serviços que dão início ao desenvolvimento de serviços baseados em IP, como audiovisual *Anytime TV* e serviços de TV em qualquer lugar. (DAVIC, 2009).

De acordo com GUILHERME e GOMARIN (2009) esta API complementa sistema de TV a cabo em todo o mundo; suas implementações aparecem com mais força na Europa, em conjunto com o DVB, e normalmente é necessário integrar com outras APIs.

3.2. API HAVi

O HAVi (*Home Audio Video Interoperability*) foi desenvolvido por um conjunto de companhias de produtos eletrônicos com o objetivo de promover a

interoperabilidade entre dispositivos de áudio e vídeo. A especificação HAVi dá suporte à interação entre dispositivos audiovisuais que compõem uma rede HAVi. De acordo com o site HAVI (2010), essa especificação é composta de APIs e *middleware* que detectam dispositivos na rede, coordena as funções desses dispositivos, coordena a instalação de aplicativos e interface do usuário de software em cada aparelho, assegurando a interoperabilidade entre dispositivos e equipamentos de áudio e vídeo independente do fabricante.

Os dispositivos HAVi compartilham seus recursos com outros dispositivos e podem também controlar e ser controlados por outros dispositivos.

As aplicações desenvolvidas por esta API suportam ambientes gráficos com funções de controle remoto, *displays* e gráficos, e podem usar componentes AWT (*Abstract Window Toolkit*) voltados para ambientes não computacionais, além de acesso ao sistema gráfico e resolução de vídeo. Ao trabalhar com imagens e seus componentes, a API HAVi oferece UI (*User Interface*) que inclui objetos com botões, caixas de diálogo, ícones, texto estático e outros. Por meio de uma classe chamada HScreen as camadas da tela podem ser configuradas. A tela da TVDI é formada por três camadas: a camada que apresenta uma imagem estática ou cor fixa no fundo, *Background layer*; a camada onde o vídeo é exibido, *Video layer*; e a camada em que as aplicações são exibidas e onde são adicionados os recursos UI, *Graphics layer*. (GUILHERME e GOMARIN, 2009). Por prover estes recursos e poder ser usado em Java foi adotado pelos sistemas de TV Digital.

3.3. API JavaTV

Desenvolvida pela *Sun Microsystems*, como extensão da plataforma Java, foi projetada para dar suporte ao desenvolvimento de aplicações para a TV Digital interativa. Esta API oferece diversas funcionalidades, tais como: fluxo de áudio e vídeo, acesso condicional, acesso aos dados nos canais de transmissão e do *Service Information*, controle do sintonizador de canais, sincronização da mídia e gerenciamento do ciclo de vida das aplicações. (GUILHERME e GOMARIN, 2009). Uma das grandes vantagens que oferece é

a portabilidade, ou seja, qualquer aplicação desenvolvida nesta plataforma pode ser executada em qualquer set-top-box, independente do hardware, desde que esteja equipado com a Máquina Virtual Java. Esta é uma das características da linguagem Java, a portabilidade, independe da plataforma – hardware, sistema operacional – bastando ter a máquina virtual Java instalada para interpretar os *bytecodes* gerados na compilação do código. A API JavaTV também independe dos sistemas de TV Digital adotados, desde que suportem JavaTV. Todos os citados neste trabalho suportam JavaTV.

Esta API está localizada no *middleware* fazendo a ponte entre o sistema operacional e as aplicações. Com a API JavaTV será possível para emissoras enviarem aplicativos interativos associados aos programas junto com a transmissão de áudio e vídeo.

É composta de várias bibliotecas que compõem a base do sistema. A especificação da API é formada pelos pacotes, listados abaixo, de acordo com a documentação da Sun para a API JavaTV, versão 1.1 (2006):

- `javax.tv.graphics`: fornece mecanismos para construção de interface usando contêiner e simples composição com componentes AWT.
- `java.tv.locator`: dá suporte à referência dos dados ou recursos que são acessíveis pela API JavaTV.
- `javax.tv.media`: fornece controle e eventos para gerir conteúdo áudio-visual em tempo real no ambiente televisivo.
- `javax.tv.net`: pacote que dá acesso aos datagramas IP transmitidos na corrente de transmissão, fluxo de *broadcast*.
- `javax.tv.service`: mecanismo de acesso a serviços de informação (SI) e APIs para o banco de dados que representam os elementos SI que ela contém. SI são informações a respeito do conteúdo dos serviços armazenados em uma base de dados.

O pacote `javax.tv.service` é composto por sub-pacotes:

- `javax.tv.service.guide`: dá suporte a aplicações do tipo EGP (Guia Eletrônico de Programação) e ao que está envolvido neste tipo de aplicação (eventos do programa, horários, programação, etc).

- `javax.tv.service.navigation`: provê mecanismos para navegação por meio dos serviços e informações de serviços de TV Digital.
- `javax.tv.service.selection`: mecanismos para descrever a forma de apresentação de um serviço.
- `javax.tv.service.transport`: descreve informações adicionais do mecanismo de transporte para um serviço de TV Digital
- `javax.tv.util`: pacote que cria e gerencia eventos de tempo.
- `javax.tv.xlet`: fornece interfaces para a gerencia das aplicações. Controla o ciclo de vida de uma aplicação para TV Digital. Este tipo de aplicação é chamado Xlet.

Para o controle de mídia – áudio e vídeo – a API JavaTV faz uso do JMF (*Java Media Framework*), definido pela SUN, para decodificar e apresentar estes tipos de conteúdo. O pacote `javax.tv.media` faz uso de extensões específicas do JMF, no caso, extensões voltadas para ambiente televisivo digital. Com isso pode-se ver que a especificação Java para a TV Digital oferece suporte ao desenvolvimento de interfaces e controle de mídia.

3.3.1. Aplicação JavaTV: XLET

Uma aplicação desenvolvida para TV Digital com a API JavaTV recebe o nome de Xlet. Os xlets são incluídos em serviços de TV Digital e são acionados por uma fonte externa, no caso, um gerente de aplicações, presente no set-top-box, que identifica o ponto de início da aplicação e executa utilizando a JVM. O xlet tem o mesmo conceito dos Applets, que são executados em páginas HTML, e dos Middlets, que são executados em dispositivos móveis usando a máquina virtual J2ME. No entanto existe uma diferença entre os xlets e applets, o gerente de aplicação pode controlar o ciclo de vida de um xlet, incluindo pausá-lo, o que não pode ser feito com uma applet.

Os xlets são independentes, possuem seus próprios recursos e não interferem na execução de outros xlets, isso possibilita que vários xlets sejam executados ao mesmo tempo em um ambiente televisivo interativo, sendo que, apenas um ficará visível e os que estiverem em estado de pausa ficarão no

estado não-visível liberando, assim, recursos para outras aplicações. A figura 3.3.1.1 ilustra uma aplicação de escolha de ângulos de câmera.

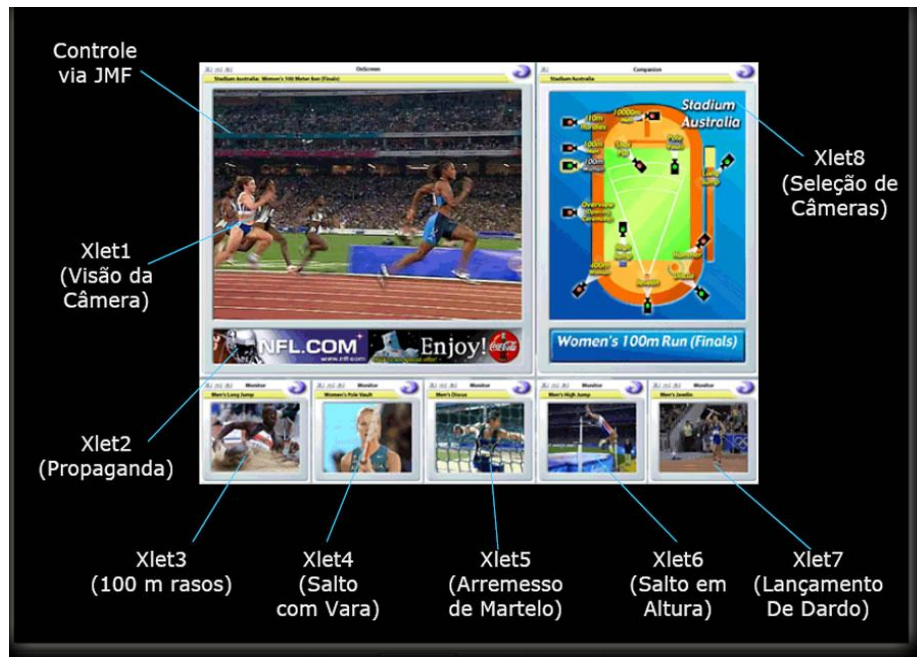


Figura 3.3.1.1: Escolha de ângulos de câmera. (CRUZ, 2009)

Como mencionado antes, o xlet pode ser controlado pelo gerente de aplicação em seus quatro estados: carregado, iniciado, pausado e destruído. Em um ambiente de TVD esse controle faz-se necessário, pois uma aplicação pode ser pausada quando o telespectador oculta a aplicação; pode ser destruída quando troca de canal. A figura 3.3.1.2 a seguir descreve os estados de um xlet.

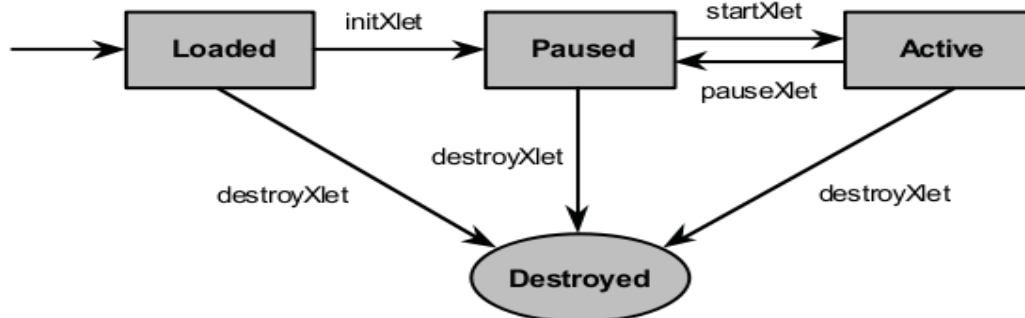


Figura 3.3.1.2: Estados de ciclo de vida de um xlet. (ABNT NBR, 2010)

De acordo com a ABNT NBR (2010) o processo do ciclo de vida do xlet ocorre da seguinte forma: quando os dados da aplicação são obtidos, o gerente de aplicação carrega a classe principal do xlet e cria uma instância chamando seu construtor, se não ocorrer exceção o xlet vai para o estado *Loaded*. Quando a xlet é iniciada pelo usuário ou por alguma aplicação o gerente chama o método *initXlet* que passa um objeto *XletContext*, que possui informações de contexto para a aplicação, incluindo carregar os arquivos que exigem maior processamento; se tiver sido carregada e instanciada com sucesso o xlet passa para o estado *Paused*. Para iniciar a execução é chamado o método *startXlet* e passa para o estado *Active* ou *Started*. Se o gerenciador chamar o método *pauseXlet* o estado da aplicação muda para *Paused*, e assim, minimizar seu consumo de recursos; se chamar o método *startXlet* a aplicação muda pra o estado *Active*. A mudança entre *Active* e *Paused* pode ocorrer várias vezes durante a vida da aplicação. Para finalizar uma aplicação, ou seja, passar o xlet para o estado *Destroyed*, o gerenciador deve chamar em qualquer estado o método *destroyXlet*, que avisa a aplicação que ela será terminada e libera todos os recursos utilizados. Depois de finalizado o xlet não pode mais voltar a ser iniciado.

A interface *Xlet* é encontrada no pacote `javax.tv.xlet` e é mostrada na figura 3.3.1.3.

```
public interface Xlet {  
  
    public void initXlet(XletContext ctx)  
        throws XletStateChangeException;  
  
    public void startXlet()  
        throws XletStateChangeException;  
  
    public void pauseXlet();  
    public void destroyXlet(boolean unconditional)  
        throws XletStateChangeException;  
}
```

Figura 3.3.1.3: Interface *Xlet*

Para obter informações do ambiente ou, também, avisar sobre a mudança de estado de um xlet, é passado um objeto *XletContext*, uma instância da classe `javax.tv.xlet.XletContext`. A figura 3.3.1.4 mostra a interface *XletContext*.

```
public interface XletContext {
    public static final String ARGS = "javax.tv.xlet.args"

    public void notifyDestroyed();
    public void notifyPaused();
    public void resumeRequest();

    public Object getXletProperty(String key);
}
```

Figura 3.3.1.4: Interface *XletContext*

Os métodos servem para notificar o gerente de aplicações a respeito das mudanças de estado. Como os próprios nomes já dão uma ideia, o método *notifyDestroyed* notifica que o xlet vai ser finalizado ou encerrado e o método *notifyPaused* notifica que o xlet entrará em pausa. O método *resumeRequest* solicita permissão para o xlet passar do estado pausado para ativado. Para obter informações do ambiente é utilizado o método *getXletProperty*. (BORDERS, 2010)

3.4. Emulador Xletview

Emulador desenvolvido em Java, e de código aberto, é usado para realizar os testes de aplicações voltadas para TV Digital no Brasil. É licenciado sob a GNU – General Public Licence e traz em seu pacote implementações de referência da API JavaTV, APIs especificada no padrão MHP, HAVi, DAVIC, DVB e bibliotecas do *PersonalJava*. Por ser desenvolvido em Java, sua execução independe de sistema operacional, sendo necessário a utilização do

Java 2 Standard Development Kit (CANDIDO JUNIOR, 2008). Para exibir vídeos utiliza o JMF 2.1.1.

A interface do Xletview apresenta um simulador de controle remoto, onde a classe AWT ou HAVI são utilizadas para capturar os eventos. A figura 3.4.1 apresenta a interface do Xletview.

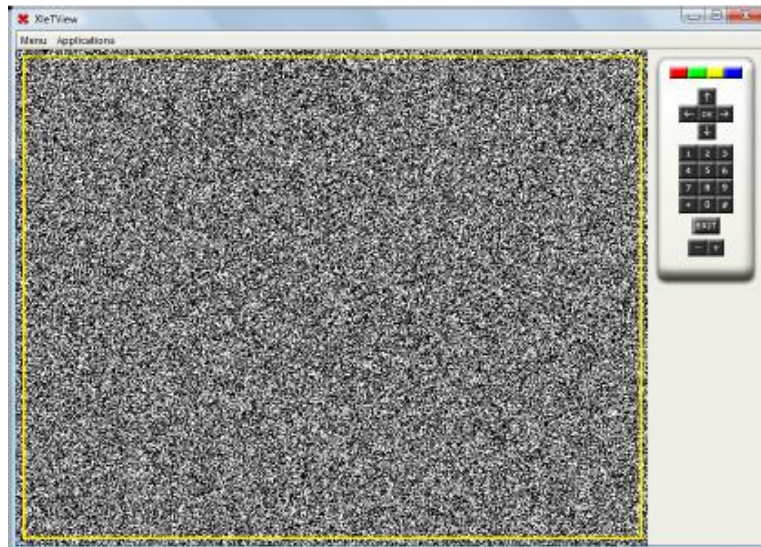


Figura 3.4.1: Interface do Xletview

No entanto, este emulador apenas testa xlets, não simula todas as funcionalidades do set-top-box, por exemplo, não simula o canal de retorno.

4. Comércio eletrônico e *t-commerce*

O avanço da tecnologia da informação facilitou a entrada dos negócios no ambiente digital. Com a popularização da internet, as empresas passaram a oferecer seus serviços via web, ou seja, oferecer seus produtos e disponibilizar meios para que o consumidor comprasse o(s) produto(s) com apenas alguns *clicks*, estando ele com um computador conectado à internet. Esse processo de compra e venda de produtos/serviços por meios eletrônicos é denominado comércio eletrônico ou *e-commerce* (O'Brien, 2002).

Para as empresas que fazem uso da internet para oferecer seus produtos, o comércio eletrônico torna-se mais do que compra e venda de produtos on-line, engloba um processo inteiro de logística, marketing, venda, entrega, atendimento e pagamento por produtos e serviços comprados por comunidades mundiais de clientes virtuais (O'Brien, 2002).

Ainda de acordo com O'Brien (2002), comércio eletrônico inclui processo de *marketing* interativo, pedidos e pagamentos na internet, acesso por *extranet* a banco de dados de estoque pelos clientes e fornecedores, acesso por *intranet* a cadastros de clientes por representantes de vendas e atendimento ao consumidor e no desenvolvimento de produtos via grupos de notícias da internet e troca de e-mail. E inclui, também, nove principais processos, como: controle de acesso e segurança; perfilização e personalização; administração de conteúdo; administradores de fluxo de trabalho; gerenciamento de busca; colaboração e comércio; sistemas de pagamentos; notificação de evento e gerenciamento de catálogos.

No entanto, para obter sucesso as empresas que utilizam sites para vender seus produtos/serviços é necessário fazer que o site seja acessado. Uma das formas de divulgar o site é por meio da televisão. Muitos canais de venda fazem marketing de seus produtos e apresentam o endereço eletrônico ou um número de telefone, para a pessoa acessar o site e comprar o produto desejado, ou fazer a compra via telefone.

Com a TV Digital interativa o telespectador poderá usar não só o site ou o telefone mais também a própria TV para comprar. Com o controle remoto ele poderá escolher o(s) produto(s), escolher a forma de pagamento e finalizar

a compra. A TV Digital interativa traz como um de seus benefícios, a possibilidade de compra de produtos e serviços oferecidos pelos programas de televisão. A ideia que se propõe é, ao estar assistindo a um programa e ver um produto de seu interesse, o telespectador clica no produto por meio do controle remoto e serão oferecidos para ele todas as informações necessárias para poder comprar o produto, se desejar. Os canais exclusivos de venda podem fazer bom uso desta tecnologia, no entanto, esse tipo de comércio eletrônico, feito por canais exclusivos de venda pela TV como Polishop, ShopTime, Shoptv, etc, é baseado em uma publicidade dirigida e maçante, que faz com que normalmente o telespectador só assista a esse tipo de canal se tiver algo que o chame a atenção, porque, normalmente, a pessoa chega a esses canais se mudar de canal em sua TV de forma seqüencial, assim, ao passar por esses canais, se tiver algum produto em destaque que chame sua atenção, talvez ele assista a apresentação do produto. No entanto se a pessoa deseja comprar um produto, um celular, por exemplo, ela vai até uma loja mais próxima ou acessa um site especializado para efetuar a compra, por isso pode tornar o sistema falho em termos de sucesso de vendas.

A revolução está em vender de forma subliminar, atingir o público em geral, como exemplo: uma mulher está assistindo a uma novela e numa cena entra uma atriz com um vestido bonito. Com um simples apertar de botão do controle remoto a telespectadora lê informações a respeito do vestido e, também, informação de como ela poderá comprar pela TV. Esse tipo de comércio é denominado *t-commerce* (*television commerce*). É mais um passo do comércio eletrônico. O *t-commerce* “soma à programação das emissoras de televisão as características básicas do *e-commerce*, inserindo-as na transmissão de conteúdos em alta qualidade de imagem e som” (ALMEIDA e CARVALHO, 2010).

As empresas de televisão devem ser as mais interessadas nos benefícios do *t-commerce*, porém, elas têm a obrigação de prover todo o suporte para que o processo do comércio pela TV tenha sucesso, ou seja, adaptar todos os processos envolvidos no *e-commerce* para o *t-commerce*, além de prover simplicidade para que o telespectador compre pela TV sem encontrar dificuldades.

Em seu artigo GHISI, LOPES e SIQUEIRA (2010), apresentam quatro modelos de *t-commerce* que serão possíveis no Brasil: Canais de Venda, Relacionado Com Programas, Publicidade Interativa e Outras Iniciativas.

No modelo Canais de Venda refere-se aos canais que oferecem produtos, mas realizam o processo por telefone ou pela internet. Esse modelo permite que o processo da venda seja feito pela TV. O modelo de *t-commerce* do tipo Relacionado Com Programas são aplicações que aparecem em programas de televisão, por exemplo, jogos de futebol, novelas, entrevistas, entre outras, por exemplo, venda da camisa do time de futebol no momento do jogo. No modelo Publicidade Interativa aplicativos específicos adicionam interatividade nos anúncios publicitários. O modelo Outras Iniciativas inclui o *t-commerce* sendo utilizados em aplicações do tipo doação de dinheiro, jogos de azar e outros.

Existe, nas duas maiores empresas de canal fechado no Brasil, NET e SKY, uma essência de *t-commerce*. Essas empresas oferecem o serviço de compra de filmes, que podem ser efetuados pelo controle remoto, o serviço *Pay-Per-View*, que é um serviço onde os telespectadores pagam por uma programação específica, por exemplo, determinado filme, onde essa programação é transmitida ao mesmo tempo para todos os que compraram. Na Net, filmes e programas em *Pay-Per-View* podem ser comprados pelo controle remoto (NET, 2010). Na SKY, o serviço Cine SKY, oferece o suporte para a compra de filmes via controle remoto (SKY, 2010).

5. Trabalhos correlatos

Sobre o estudo de TV Digital e interatividade há algumas contribuições sobre os aspectos relevantes no que diz respeito à tecnologia envolvida no sistema de TVD e no desenvolvimento das aplicações interativas.

Para citar alguns, pode-se notar o trabalho de CANDIDO JUNIOR (2008), o qual mostra os conceitos básicos e definições sobre esta tecnologia, os padrões de TV Digital existentes, incluindo o brasileiro, apresenta algumas ferramentas necessárias para o desenvolvimento de uma aplicação para a TV Digital e utiliza a API JavaTV para desenvolver uma aplicação interativa que simula uma votação pela TV, onde o telespectador participa de uma enquete e a contagem dos votos são registrados na emissora.

O trabalho de GUILHERME e GOMARIN (2009), também apresenta os conceitos sobre TVD, abordando o impacto que pode causar nos hábitos das pessoas. Descreve a arquitetura envolvida nesta tecnologia e as tecnologias envolvidas no desenvolvimento das aplicações interativas. A aplicação apresentada neste trabalho simula um ambiente onde o telespectador pode agendar com a emissora a exibição de vídeos disponibilizados pela mesma, deixando, assim, de depender da programação contínua da emissora. Neste trabalho a aplicação foi demonstrada usando a API JavaTV e simulada no emulador xletview.

Uma das fontes estudadas que se relacionam com a aplicação de comércio eletrônico pela TV, o *t-commerce*, foi o trabalho de GHISI, LOPES e SIQUEIRA (2010), que apresenta um estudo sobre quatro modelos de *t-commerce* que serão possíveis de serem aplicados com a interatividade da TV Digital. Este trabalho abordou aspectos do middleware Ginga e sua arquitetura.

O artigo de SANTOS (2007) aborda o conceito de *t-commerce*, apresentando, também, o funcionamento e a arquitetura da TV Digital e os padrões existentes e a API JavaTV. A aplicação de *t-commerce* apresentada é voltada para um sistema de leilão interativo, em que o telespectador poderá adquirir produtos por meio de lances. O emulador xletview é usado para executar a aplicação, simulando o ambiente televisivo.

6. TV-Sellcenter: uma aplicação de *t-commerce*

A partir deste ponto a ênfase será dada ao projeto que será desenvolvido neste trabalho. O aplicativo tem como objetivo mostrar o potencial da interatividade na TV Digital, baseando-se no comércio eletrônico pela TV, o *t-commerce*.

O aplicativo foi desenvolvido em linguagem Java, com a utilização das APIs JavaTV, HAVi, Davic. Foi utilizado a IDE (*Integrated Development Environment*) NetBeans 6.0.1 na plataforma *Windows* e o simulador Xletview. Os vídeos estão no formato AVI com codec CINEPAC e para serem exibidos é necessário a configuração do arquivo *channels* na pasta *config* do xletview, colocando o endereço do vídeo na linha “Media” do arquivo *channels.xml*.

A aplicação apresentada é apenas uma simulação de um ambiente televisivo digital com sua interatividade. Em um ambiente real há a necessidade de um canal de retorno para que haja uma comunicação real do usuário com a emissora. No entanto o ambiente nos dois modelos apresentados funciona de forma exatamente iguais com relação à estrutura apresentada nos capítulos anteriores sobre o funcionamento dos xlets. De forma simulada pelo emulador xletview o funcionamento acontece da seguinte forma: o *middleware*, depois de receber o pacote de dados enviados pela emissora de TV, carrega o xlet e inicia a aplicação; internamente quem faz esse trabalho é o gerente de aplicações, que recebe uma notificação da chegada de uma aplicação e a partir deste momento carrega a classe principal do xlet, e controla o ciclo da vida da aplicação. (Figura 6.1).

Nos modelos apresentados as classes principais são as classes que implementam a interface Xlet. Abaixo é mostrado o código da aplicação em cada modelo. A figura 6.2 exhibe o código da implementação da classe Xlet no modelo Canal de Vendas e a figura 6.3 exhibe a classe do modelo Relacionado Com Programa. Em cada código o ambiente em que os componentes são inseridos é preparado no método *initXlet()* (linha 8) junto com o contexto, logo em seguida, na linha 13, por meio do método *startXlet()* a aplicação é iniciada instanciando a classe que constrói a primeira tela da aplicação; no primeiro código o método da classe BtInicio (linha 15) exhibe o componente “Comprar

Produto” mostrado na figura 6.4; no segundo código o método da classe MatrixInit (linha 15) exibe o componente “Iniciar Aplicação” mostrado na figura 6.9. A partir da tela inicial o telespectador navega na aplicação por meio do controle remoto, basicamente em uma estrutura parecida com páginas *web*, ou seja, os botões do controle remoto servem como *links* para a tela seguinte. No Apêndice serão mostradas as classes que compõe a aplicação.

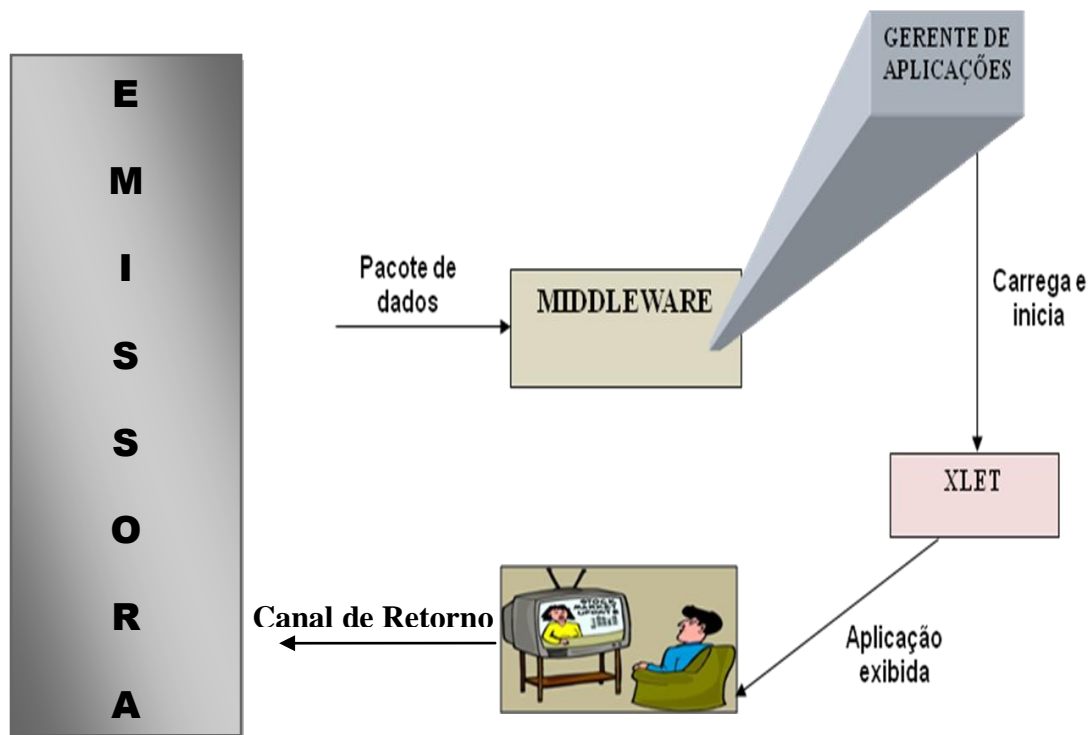


Figura 6.1: Arquitetura do funcionamento da camada de aplicação.

As telas foram construídas com o uso das bibliotecas Java.awt, Havi e DVB que permite a construção de contêineres, para inclusão de imagens e textos estáticos, e a captura e tratamento de eventos sinalizados pelo controle remoto.

```

1. import javax.tv.xlet.*;
2. import org.havi.ui.*;
3.     public class ShopTV implements Xlet{
4.         private XletContext contexto;
5.         private HScene cena;
6.         public ShopTV(){
7.             }
8.         public void initXlet(XletContext context) throws XletStateChangeException {
9.             contexto = context;
10.            cena = HSceneFactory.getInstance().getFullScreenScene
11.                (HScreen.getDefaultHScreen().getDefaultHGraphicsDevice());
12.        }
13.        public void startXlet() throws XletStateChangeException {
14.            BtInicio bt = new BtInicio(cena);
15.            bt.initBt();
16.        }
17.        public void pauseXlet() {
18.        }
19.        public void destroyXlet(boolean arg0) throws XletStateChangeException {
20.            contexto.notifyDestroyed();
21.        }
22.    }

```

Figura 6.2: Código da implementação da interface Xlet – Modelo Canal de Vendas.

```

1. import javax.tv.xlet.*;
2. import org.havi.ui.*;
3. public class Matrix implements Xlet {
4.     private XletContext contexto;
5.     private HScene cena;
6.     public Matrix(){
7.     }
8.     public void initXlet(XletContext context) throws XletStateChangeException {
9.         contexto = context;
10.        cena = HSceneFactory.getInstance().getFullScreenScene
11.            (HScreen.getDefaultHScreen().getDefaultHGraphicsDevice());
12.    }
13.    public void startXlet() throws XletStateChangeException {
14.        MatrixInit mt = new MatrixInit(cena);
15.        mt.init();
16.    }
17.    public void destroyXlet(boolean arg) throws XletStateChangeException {
18.        if (arg == true)
19.            System.out.print("Xlet destruido");
20.        contexto.notifyDestroyed();
21.    }
22.    public void pauseXlet() {
23.    }
24. }

```

Figura 6.3: Código da implementação da interface Xlet – Modelo Relacionado Com Programas.

O aplicativo proposto, denominado TV-Sellcenter, está dividido em duas partes, demonstrando dois modelos de *t-commerce* apresentados no capítulo quatro: o modelo Canal de Vendas e o modelo Relacionado Com Programas. O primeiro consiste em vendas de produtos a partir de canais de vendas. De acordo com o produto que está sendo apresentado é enviado uma mensagem convidando o telespectador a comprar o produto ou exibir uma lista com outros produtos oferecidos pelo canal. Este modelo pode ser muito utilizado pelos canais de televenda, porque a TV poderá ser mais um canal dessas empresas para o telespectador efetuar a compra. No segundo modelo, a aplicação aparecerá no momento da exibição de um filme, a mensagem convidará o telespectador a comprar um acessório que o personagem do filme usa, por exemplo, óculos; o telespectador poderá também comprar outros acessórios que fazem parte do filme. Este tipo de modelo de *t-commerce* pode ser bastante utilizado pelas emissoras, pois podem se aproveitar da atenção e das emoções que o telespectador dar ao programa para vender o produto. A estrutura da aplicação será explicada em mais detalhes, começando com o primeiro modelo.

Em um canal de vendas, como exemplo foi usado o BestShopTV.com, enquanto a programação é exibida, ou seja, a apresentação de um produto é feita, é mostrado na tela uma mensagem ao telespectador sugerindo que compre o produto (figura 6.4).

Se o telespectador clicar *EXIT* no controle remoto a aplicação é encerrada, se clicar no botão *OK* é exibida as informações sobre o produto, no caso o celular, e mais duas opções de escolha: ver mais produtos oferecidos e comprar o produto (Figura 6.5).

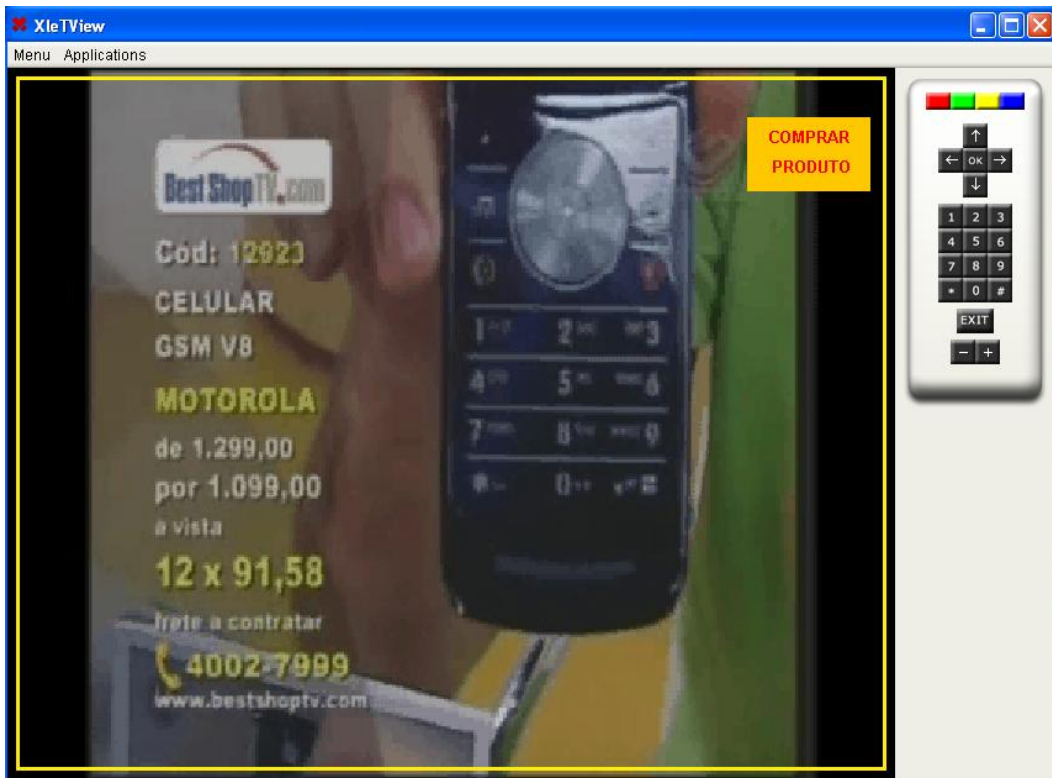


Figura 6.4: Tela inicial modelo Canal de Vendas

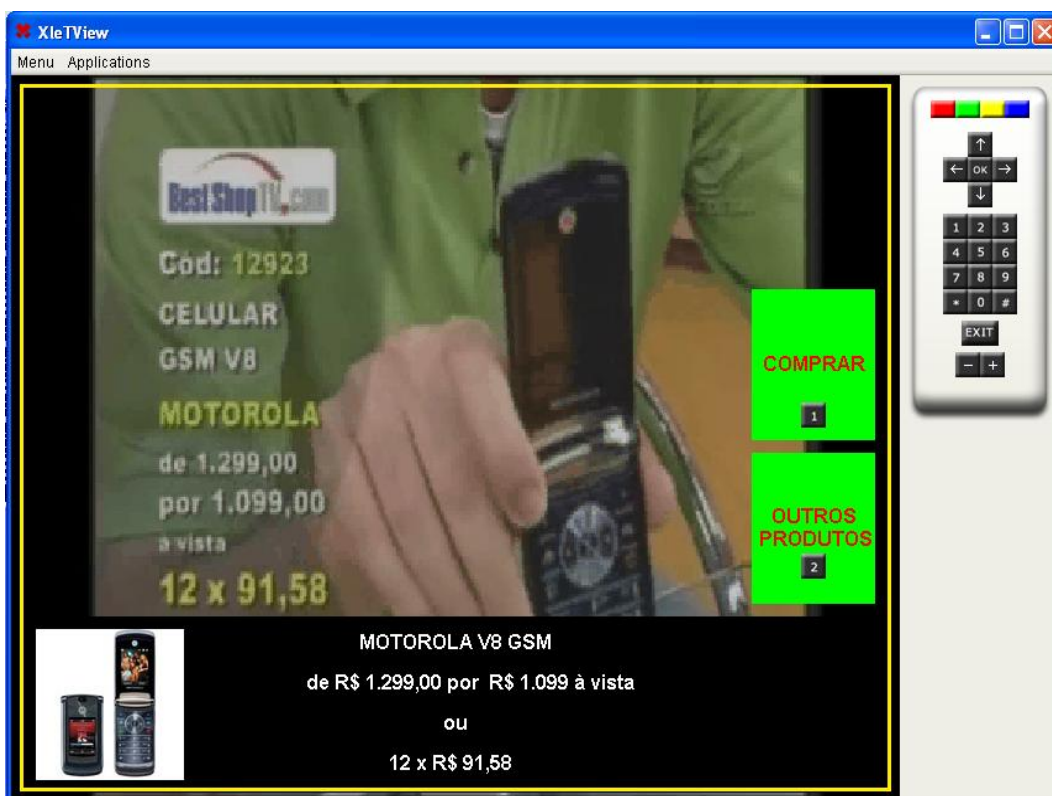


Figura 6.5: Tela apresentando informações do produto.

Se a opção 1 for escolhida, clicando o botão número 1, é mostrando uma lista com outros produtos, imagens e informação (Figura 6.6), ficando a critério do usuário ver os outros produtos por clicar no número correspondente. A figura 6.7 exemplifica a tela caso a escolha seja do produto número 1. As outras telas dos outros produtos seguem o mesmo padrão de apresentação. Caso o telespectador decida comprar, escolhendo a opção 1, é exibido uma tela mostrando as opções de cartão de crédito para pagamento (Figura 6.8). Escolhendo o tipo de cartão, o processo de compra seguiria por inserir o número do cartão e a escolha das parcelas, finalizando o processo de compra.

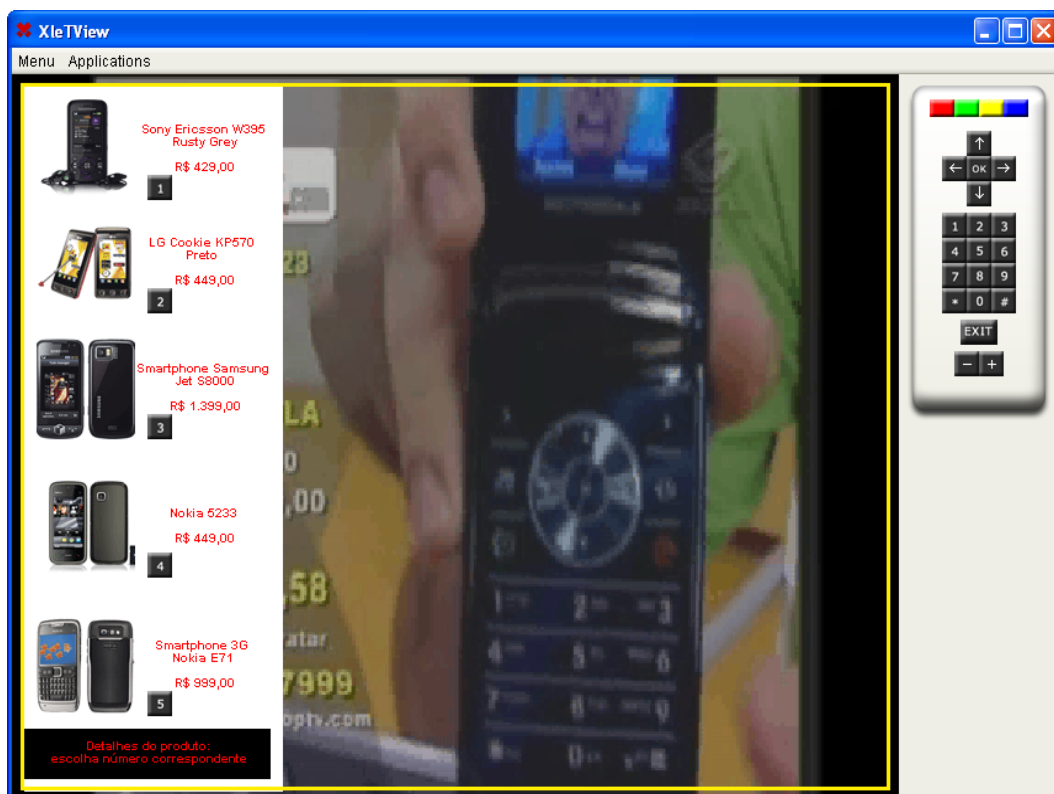


Figura 6.6: Tela apresentando outros produtos.

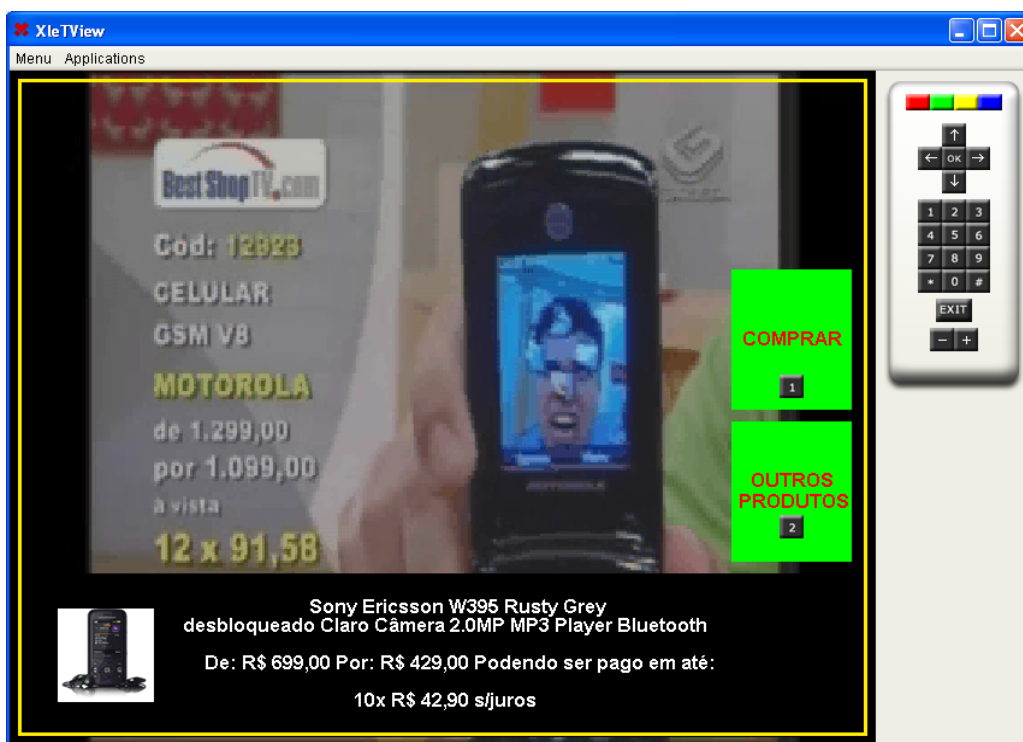


Figura 6.7: Tela apresentando o produto número 1.

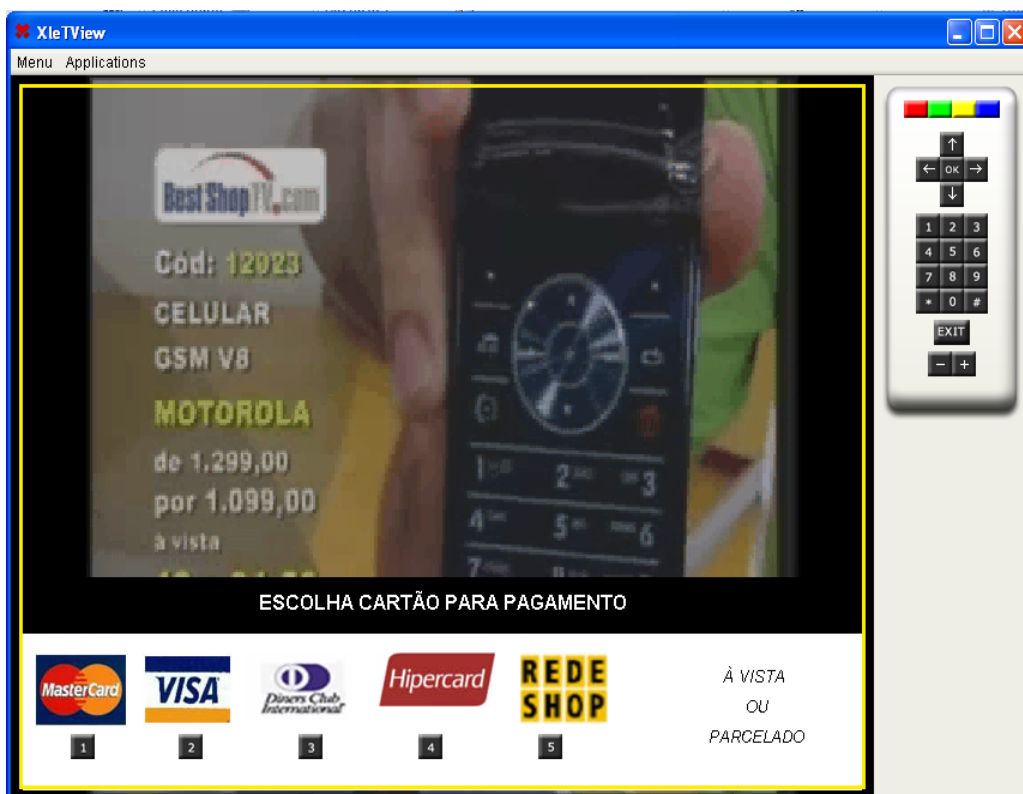


Figura 6.8: Tela apresentando cartões para pagamento.

A aplicação do segundo modelo de *t-commerce* consiste em apresentar um produto à medida que um filme está sendo exibido. Simulando que a pessoa esteja assistindo a um filme, no caso, o filme escolhido foi Matrix, é exibida na tela uma mensagem incentivando o telespectador a iniciar a aplicação que a emissora apresenta (Figura 6.9). Se o telespectador clicar *OK*, será apresentada a publicidade do produto que faz parte do acessório dos personagens do filme. Neste caso, será apresentado o óculos que os personagens do filme usam (Figura 6.10). Escolhendo ver o produto por clicar *OK*, a tela seguinte exibe informações sobre o produto e opções sobre comprar o produto ou ver outros produtos relacionados ao filme (Figura 6.11). Se a opção escolhida for a de comprar o produto, a próxima tela a ser exibida são as opções de cartão de crédito (Figura 6.12) a serem escolhidas e após esta etapa ocorreria o processo de inserir o número do cartão e a o tipo de pagamento e finalizando o processo de compra. Caso o telespectador decida ver os outros produtos relacionados então é exibida a tela com os produtos relacionados (Figura 6.13). Qualquer opção escolhida leva à tela que exibe informações do produto escolhido (Figura 6.14 e Figura 6.15), escolhendo comprar, o processo é o mesmo explicado anteriormente. Há, também, a possibilidade de voltar a ver o primeiro produto apresentado. A qualquer momento a aplicação pode ser encerrada por clicar *EXIT* no controle remoto.

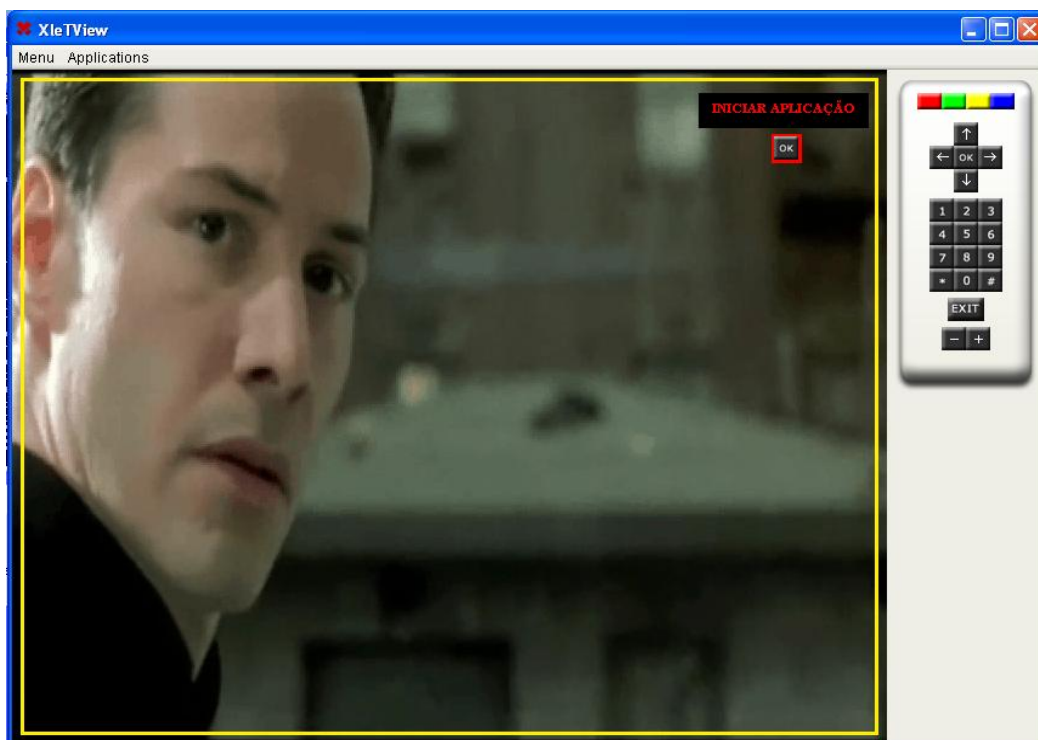


Figura 6.9: Tela inicial com a mensagem que indica uma aplicação na programação.

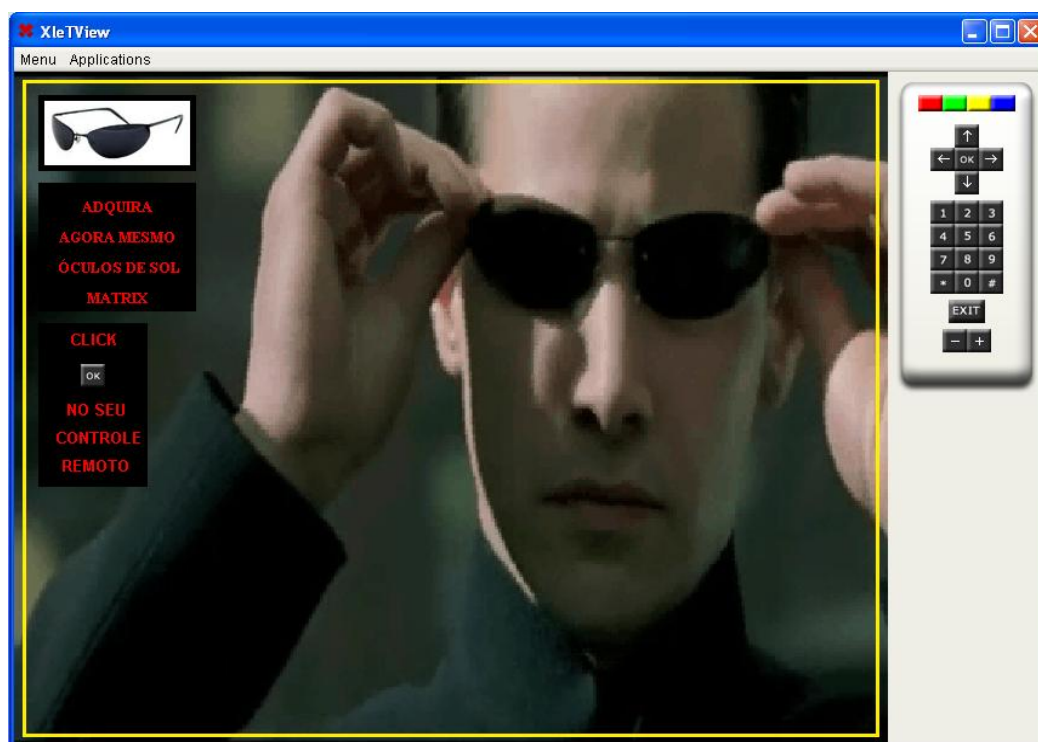


Figura 6.10: Tela apresentando produto relacionado ao filme.



Figura 6.11: Tela apresentando informações do produto.

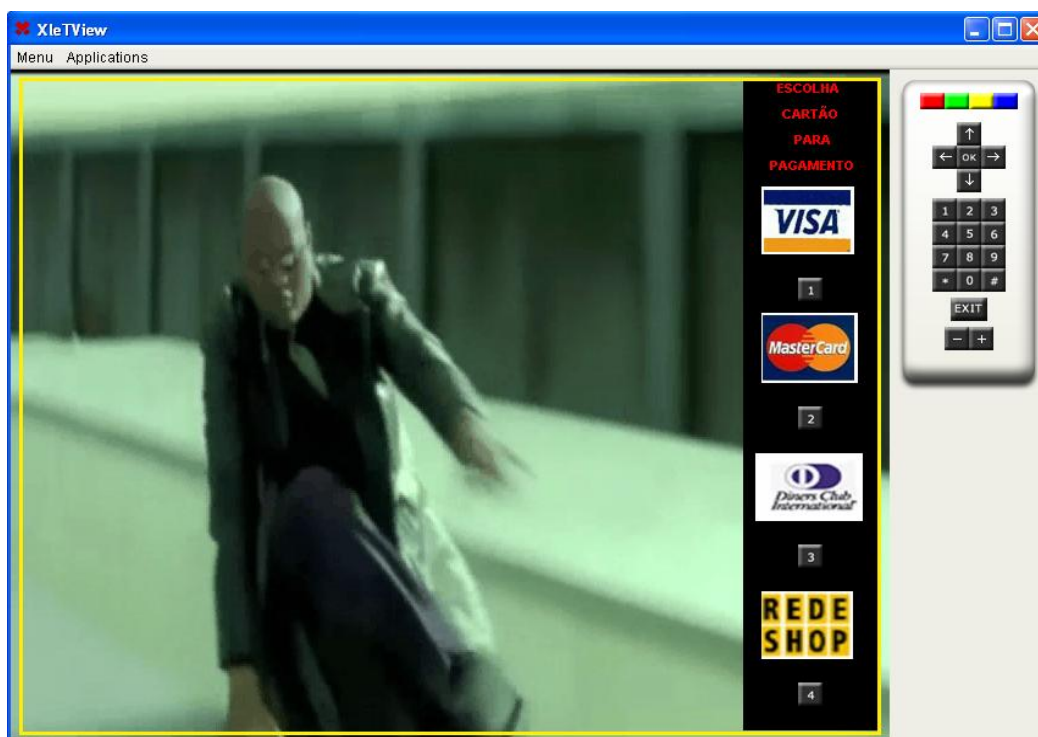


Figura 6.12: Tela apresentando cartões para pagamento.

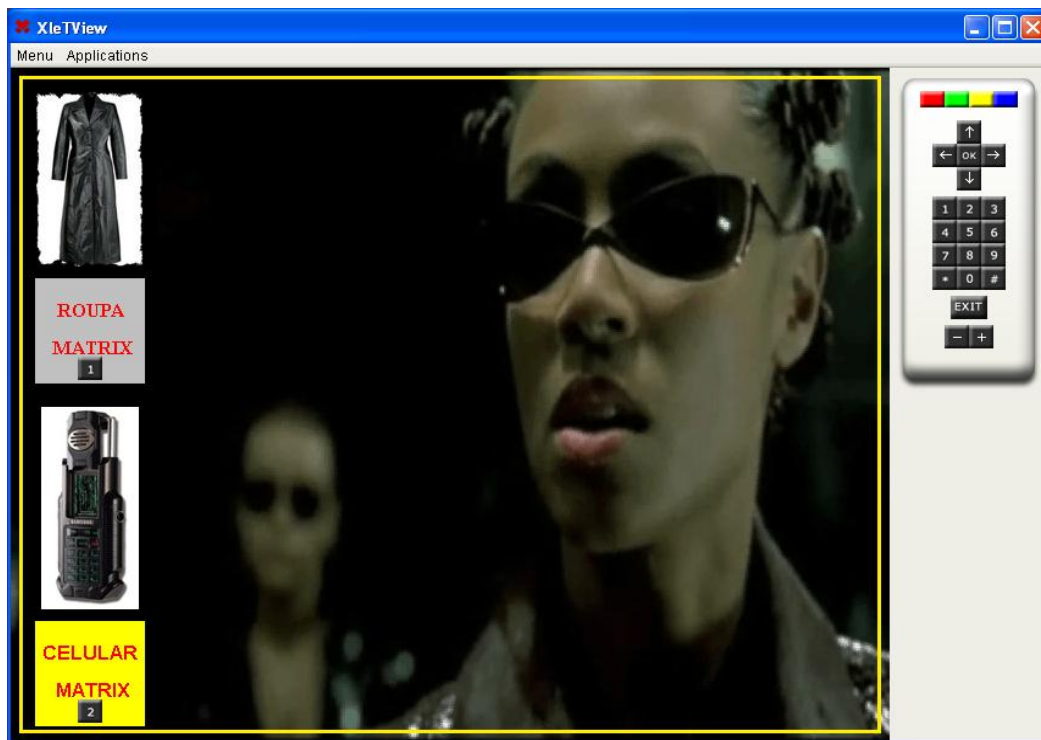


Figura 6.13: Tela apresentando outros produtos relacionados.

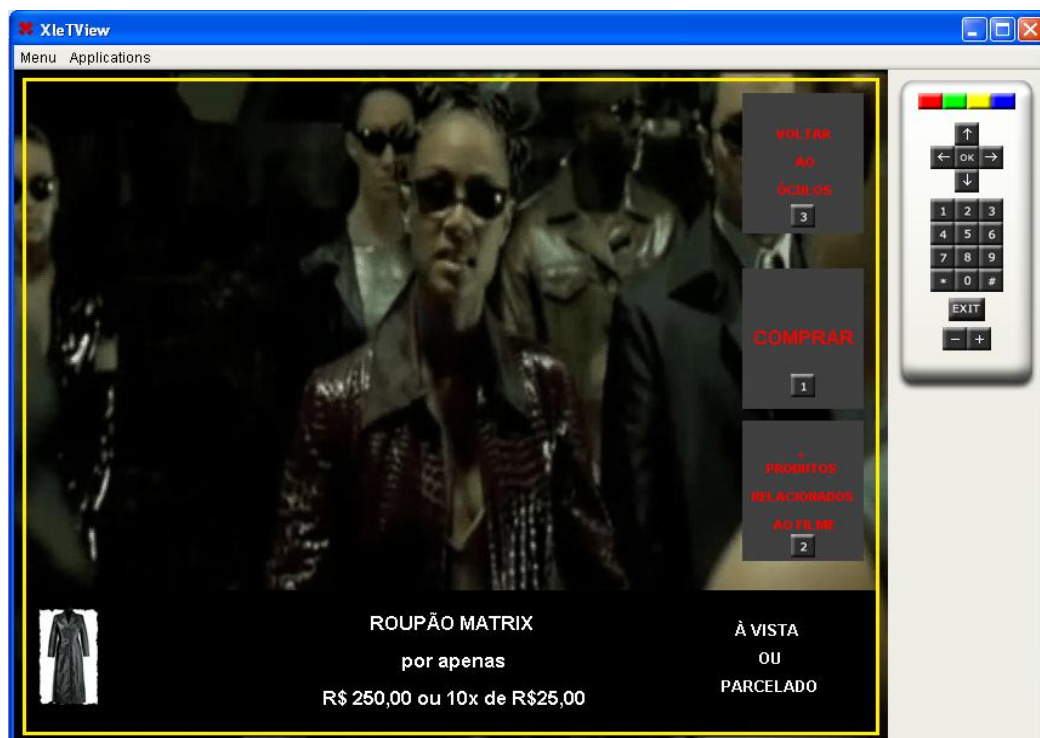


Figura 6.14: Tela apresentando produto 1 – Roupas Matrix.

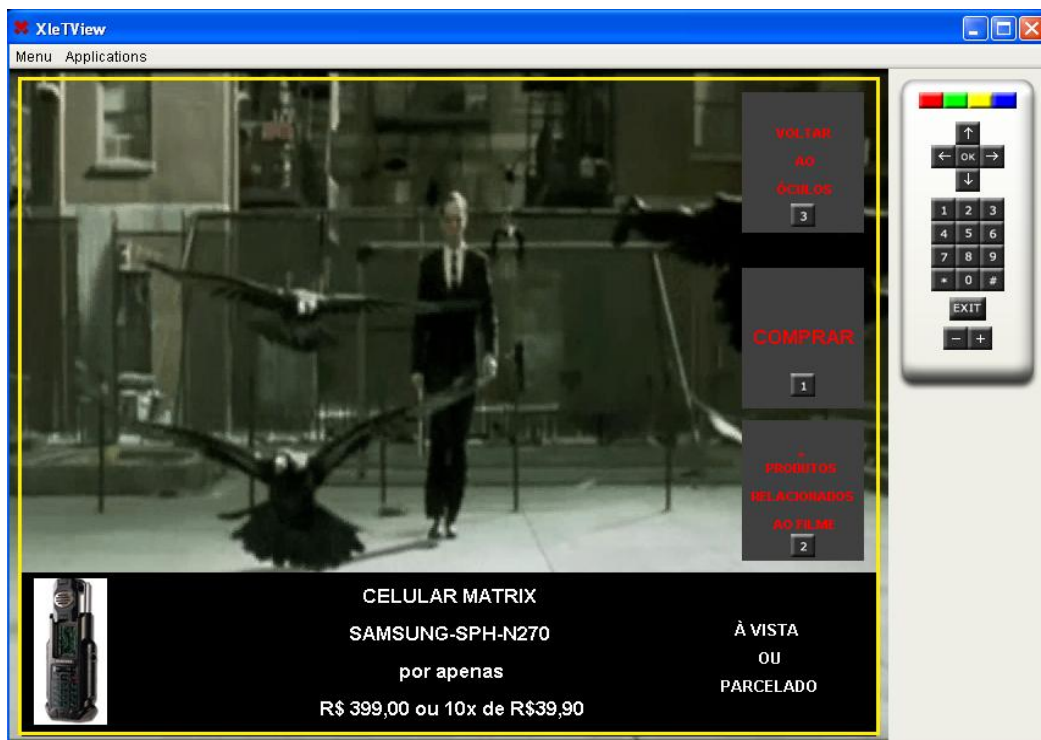


Figura 6.15: Tela apresentando produto 2 – Celular Matrix.

7. Conclusões

Por prover suporte à interatividade, a TV Digital muda a forma de o telespectador assistir televisão, deixando de ser um usuário passivo da mesma e passando a ser um participante ativo com o conteúdo televisivo, transmitido pela emissora. Com isso, vários tipos de aplicações poderão ser desenvolvidas para a TV Digital, incluindo aplicações para o comércio eletrônico, que atualmente é feito pela internet, mas poderá ser feito pela TV, portanto:

- A aplicação desenvolvida neste projeto forneceu uma ideia de como será este tipo de comércio em um futuro próximo, onde o telespectador poderá interagir com a programação efetuando a compra do produto apresentado pela emissora, sendo assim, a TV se torna mais meio eletrônico de compras, além dos já existentes, internet e telefone.
- Pôde-se ver que é possível desenvolver aplicações para esse sistema de TV e explorar a sua potencialidade e interatividade. A API JavaTV em conjunto com outras ferramentas fornece subsídios para o desenvolvimento de aplicações procedurais e, teoricamente, esta API é compatível com o sistema brasileiro Ginga-J, visto que neste trabalho, por falta de recursos, não foi possível testar esta aplicação em um ambiente real, ou seja, em uma TV com o middleware Ginga instalado, mas, teoricamente, qualquer aplicação desenvolvida com JavaTV será executada com o Ginga-J ou com outros sistemas compatíveis.

O padrão de TV Digital brasileiro já é uma realidade, a maioria dos fabricantes de TV já incorporou o DTV nos conversores digitais embutidos e fabricantes como a LG já incorporou o Ginga em seus televisores; o que se espera agora é aprimorar o desenvolvimento de software e encontrar fortes investidores que acreditem nesta nova e promissora forma de fechar negócios.

O resultado obtido com a experimentação deve ser entendido como promissor, haja vista que foi possível desenvolver uma aplicação que explore a ideia de *t-commerce* desenvolvida neste trabalho, entretanto, diversos outros

critérios devem ser levados em consideração para poder converter a ideia em fato, ou seja, converter a teoria em prática.

Assim sendo, como linhas futuras de pesquisa, fica registrada a necessidade de pesquisar, estudar e analisar os diferentes aspectos que envolvem a:

- Usabilidade, ou seja, a facilidade do telespectador interagir com a televisão; sendo que é um processo não muito rápido, o processo de compra tem que ser fácil, pois a pessoa pode não efetuar a compra se não for fácil para ele fazer isso.
- Interface, pois uma interface amigável e interessante, que chame a atenção do telespectador, pode ser um fator importante para se vender pela TV.
- Aceitação deste tipo de serviço, pois o comércio eletrônico via internet já está bem estabelecido, existe também a compra via telefone e mesmo assim as pessoas vão até as lojas físicas fazer compras, por outro lado a televisão é uma mídia que está na maioria das casas dos brasileiros, sendo assim, o estudo sobre a aceitação do *t-commerce* poderia mostrar se este serviço interativo terá sucesso ou não.

Referências

ALMEIDA, W. F.; CARVALHO, J. M. **T-commerce e modelo de negócio: o comércio eletrônico televisivo nos canais do sistema brasileiro do agronegócio**. Celacom 2010: Televisão na América Latina, 2010.

ARIB, **Association of Radio Industries and Businesses**. Disponível em <<http://www.arib.or.jp/english/html/overview/doc/6-STD-B23v1>> Acessado em: 25/05/2010

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 15602-1**. Televisão digital terrestre — Codificação de vídeo, áudio e multiplexação. Parte 1: Codificação de vídeo, 2007.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 15606-4**. Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital. Parte 4: Giga-J - Ambiente para a execução de aplicações procedurais, 2010.

ATSC, **Advanced Television Systems Committee**. Disponível em <<http://www.atsc.org/>> Acessado em 25/05/2010.

BORDERS, T. W. **Uma introdução ao MHP**. Disponível em: <<http://www.mhp-interactive.org/tutorials/mhp>> Acessado em 23 maio 2010.

BRACKMANN, C. P. **Sistema brasileiro de TV digital**. Pelotas, 2008. Universidade Católica de Pelotas.

CANDIDO JUNIOR, E. **Utilização da API JavaTV para o desenvolvimento de aplicações para a TV digital interativa**. São Paulo, 2008. Monografia (Especialização em Desenvolvimento de Sistemas para Ambientes baseados em Tecnologia Java) – Universidade do Oeste Paulista.

CRUZ, D. A. **Tecnologia, Computação e Internet**. Disponível em: <<http://danielcruz733.wordpress.com/2009/08/04/a-api-javatv/>> Acessado em: 30 maio 2010.

DAVIC. **Digital Audio Visual Council**. Disponível em: <<http://www.davic.org/>> Acessado em: 02 jun. 2010.

DIBEG. **Digital Broadcasting Experts Group**. Disponível em: <<http://www.dibeg.org/>> Acessado em: 25 maio 2010.

DTV. **Site Oficial da TV Digital Brasileira**. Disponível em: <<http://www.dtv.org.br/>> Acessado em: 17 maio 2010.

GATIS, I. de A. L. **Um middleware para construção de aplicações de TV digital distribuídas baseadas no modelo P2P**. Recife, 2006. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco.

GHISI, B. C.; LOPES, G. F.; SIQUEIRA, F. **Conceptual models for t-commerce in Brazil** Workshop on Interactive Digital TV in Emergent Countries - EuroITV2010

- GINGA. **TV interativa se com ginga.** Disponível em: <<http://www.ginga.org.br/>> Acessado em: 30 maio 2010.
- GINGA-J, M. **Ginga CDN. Code Development Network.** Disponível em: <<http://dev.openginga.org/>> Acessado em: 30 maio 2010.
- GINGA NCL. **Ginga NCL.** Disponível em: <<http://www.gingancl.org.br/>> Acessado em: 30 maio 2010.
- GUILHERME, J. F.; GOMARIN, W. C. **Desenvolvimento de um aplicativo para TV digital usando as bibliotecas da JavaTV.** Franca, 2009. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade de Franca.
- HAVI. **Home Audio Video Interoperability.** Disponível em: <<http://www.havi.org/>> Acessado em: 02 jun. 2010.
- IDG NOW. **IDG NOW! Tecnologia em primeiro lugar.** Disponível em <<http://idgnow.uol.com.br>> Acessado em: 30 maio 2010.
- MONTEZ, C.; BECKER, V. **TV Digital Interativa: conceitos, desafios e perspectivas para o Brasil.** Florianópolis: Ed. da UFSC, 2005. 2ª edição.
- NET. **NET – O mundo é dos nets.** Disponível em: <<http://netcombo.globo.com/>> Acessado em 10 jun 2010.
- O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet.** Ed. Saraiva, 2002. 9º edição.
- PROTEVE. **Mídia Eletrônica Profissional.** Disponível em: <<http://www.proteve.net/modulacao.html>> Acessado em 22 jun 2010.
- SANTOS, E. R. dos, **T-commerce, adaptação de serviços de comércio eletrônico para TV digital.** Canoas, 2007. Artigo - Universidade Luterana do Brasil.
- SKY. **SKY HDTV é isso.** Disponível em: <<http://www.sky.com.br/sonasky/interatividade>> Acessado em 10 jun 2010.
- SOUZA FILHO, G. L. de; LEITE, Luiz E. C.; BATISTA, C. E. C. F. **Ginga-J: The Procedural Middleware for the Brazilian Digital TV System.** In: _____ Journal of the Brazilian Computer Society. No. 4, Vol. 13. p.47-56. ISSN: 0104-6500. Porto Alegre, RS, 2007.
- SUN, Microsystems. **Java TV API 1.1 (JSR-927).** Disponível em: <<http://java.sun.com/javame/reference/apis/jsr927/index.html>> Acessado em: 02 jun. 2010.
- TANENBAUM, A. S. **Redes de computadores.** Ed. Campus, 2003. 4º edição.
- TVMAGAZINE. **TV Magazine.** Disponível em:<www.tvmagazine.com.br/blogs/post.asp?ID=98> Acessado em: 17 maio 2010.

Apêndice

O aplicativo TVSellCenter é composto de dois módulos, representando dois modelos de *t-commerce* que poderão ser utilizados na TV Digital, sendo que, são apresentados como xlets distintos. O primeiro é composto de seis classes iniciada pela classe principal ShopTv que inicia a aplicação, como mostra na figura 8.1 abaixo:

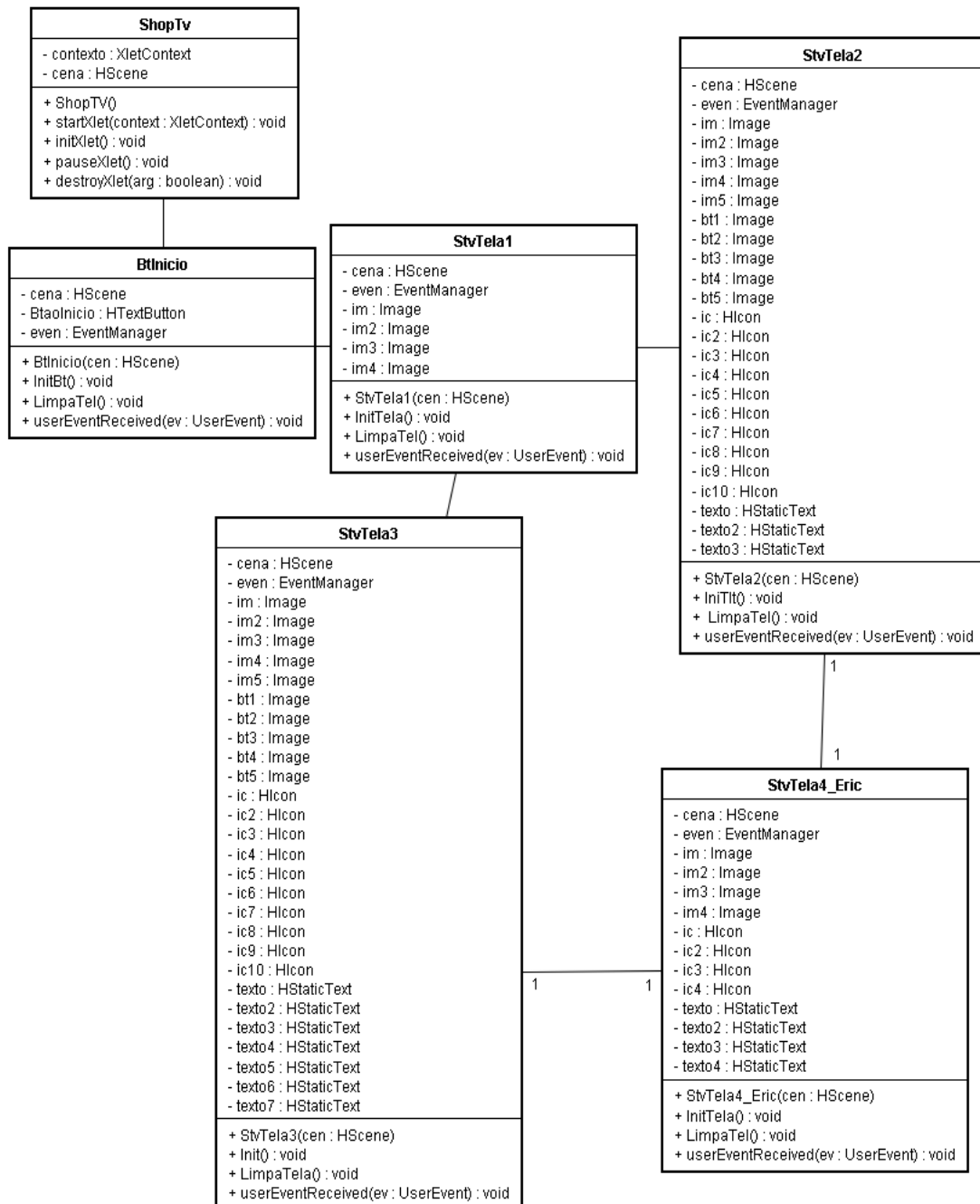


Figura 8.1: Diagrama de classes - ShopTv

A classe principal é a ShopTv, que implementa a interface xlet nela se encontram os métodos que controlam o ciclo de vida de um xlet. Neste caso específico o método initXlet instancia a classe BtInicio, essa classe é responsável por exibir ao telespectador o aviso de “Comprar produto”. A classe StvTela1 é responsável por exibir informações a respeito do produto apresentado no programa e as opções para a escolha. A classe StvTela2 exibe informações a respeito da forma de pagamento, ela é instanciada no momento que o telespectador escolhe comprar o produto. A classe StvTela3 é responsável por exibir os outros produtos apresentados para que o usuário escolha outro produto, se desejar, e a classe StvTela4_Eric exibe informações do primeiro produto apresentado na tela anterior. Todas as classes, com exceção da classe principal, capturam as ações do telespectador ao fazer suas escolhas.

O segundo aplicativo é composto de oito classes seguindo a mesma lógica mostrada anteriormente, abaixo, figura 8.2, é mostrado o diagrama de classe do segundo modelo. Ele é iniciado com a classe que implementa os métodos da interface Xlet, essa classe principal foi denominada Matrix, que prepara o ambiente e instancia a classe MatrixInit por meio do método initXlet, a classe instanciada é responsável por exibir a informação que uma aplicação pode ser ativa. A classe MatrixTel1 exibe o comercial do produto que faz parte do programa em exibição; a classe MatrixTel2 exibe informações a respeito do produto e opções para o telespectador navegar. As classes seguintes exibem informações sobre a forma de pagamento, classe MatrixTel3, é responsável por mostrar outras opções de produtos relacionados ao programa, classe MatrixTel4, e por mostrar as informações a respeito dos outros itens apresentados, classe MatrixTel5 e classe MatrixTel6. Com exceção da classe principal todas as outras capturam as ações do telespectador ao navegar nas telas apresentadas.

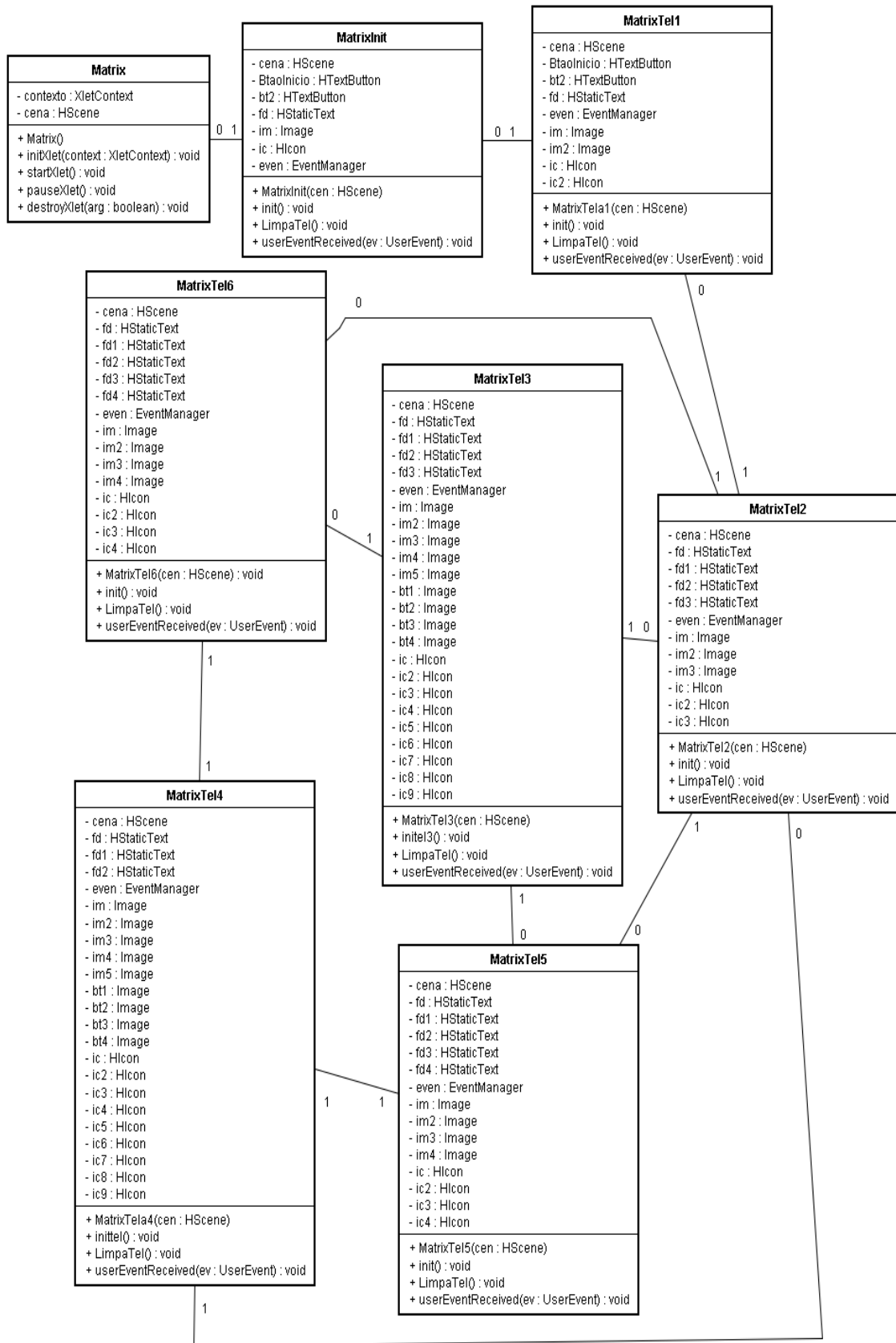


Figura 8.2: Diagrama de classes - Matrix