

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSA PARA O DESENVOLVIMENTO
DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

MARCEL BEZERRA DE AGUIAR

**DESENVOLVIMENTO PARA TV DIGITAL UTILIZANDO
A TECNOLOGIA JAVA**

**NATAL/RN
2008**

Catálogo de Publicações – Biblioteca da FARN
Setor de Processos Técnicos

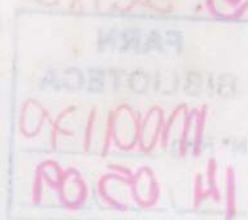
MARCEL BEZERRA DE AGUIAR

**DESENVOLVIMENTO PARA TV DIGITAL UTILIZANDO
A TECNOLOGIA JAVA**

Monografia apresentada ao Curso de Graduação de bacharelado em Sistemas de Informação pela Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito obrigatório para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:
Gleydson Lima, Msc.

NATAL/RN
2008



Aguiar, Marcel Bezerra de

Desenvolvimento para TV digital utilizando a tecnologia java / Marcel Bezerra de Aguiar. – Natal: FARN, 2009.

51f.

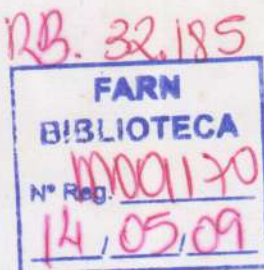
Orientador: Gleydson Lima (Msc.)

Monografia (Graduação em Sistemas de Informação) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Sistemas de Informação – Monografia. 2. TV Digital – Monografia. 3. Plataforma Java – Monografia. 4. Título.

RN/FARN/BC

CDU 621.397



621.397
A23d
2008
ex. 01

NATALIAN
2008

MARCEL BEZERRA DE AGUIAR

DESENVOLVIMENTO PARA TV DIGITAL UTILIZANDO
A TECNOLOGIA JAVA

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte, aprovada em ____ de ____ de ____, pela
banca examinadora composta pelos seguintes membros:

Gleydson de Azevedo F. Lima

Gleydson Lima, Msc.

Orientador

Edgard de Faria Correa, Dr.

Membro examinador

Joseane Alves Pinheiro.

Membro examinador

AGRADECIMENTOS

Agradeço principalmente a Deus por me mostrar o caminho certo, pela força para superar os obstáculos, vencer os desafios e ir além para que eu pudesse continuar.

A meus pais pela educação que me passaram durante toda minha vida, pela fé e pela força para que eu terminasse mais essa etapa de minha vida e que agora são os três filhos com formação superior.

A minha filha Eduarda, fonte de inspiração e motivo pelo qual tenho a cada dia, mais e mais forças para trabalhar, estudar e conquistar meus objetivos.

A minhas avós que apesar das brigas sempre estiveram ao meu lado e sempre me ajudaram.

A minha Avó Dodá que sempre torceu por mim e que esteve ali, para me ajudar, nos momentos difíceis.

A minha tia Dantas (Bela), por sempre me apoiar, ajudar e por ter fi na mim.

A Livia Lopes Matos, por me ajudar a seguir na frente e não desistir dos meus objetivos durante esses quatro anos de curso.

A todos os meus amigos, principalmente José Lopes da Silva Neto, pelo apoio e ajuda.

Aos meus colegas de classe que me ajudaram quando não pude estar presente nas aulas, pelo ajuda nos trabalhos, aulas pelo Google Groups etc.

A todos os professores pela paciência com que sempre me recebeu e pela ajuda nas atividades e aulas que foram necessárias.

A todos as pessoas que me ajudaram direta ou indiretamente neste curso.

Dedico este trabalho monográfico aos meus pais Baltazar e Jurema,
as minhas irmãs Maria Luiza e Milla, a minha avó Dodá
e principalmente a minha querida filha Eduarda,
que sempre me incentivaram e continuam
torcendo por meu sucesso profissional.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me mostrar o caminho certo, pela força para superar os obstáculos, vencer os desafios e saúde para que eu pudesse continuar.

A meus pais pela educação que me passaram durante toda minha vida, pela torcida para que eu terminasse mais essa etapa de minha vida e que agora são os três filhos com formação superior.

À minha filha Eduarda, fonte de inspiração e motivo pelo qual tenho a cada dia, mais e mais forças para continuar a trabalhar, estudar e conquistar meus objetivos.

As minhas irmãs que apesar das brigas sempre estiveram ao meu lado e sempre me ajudaram.

À minha Avó Dodá que sempre torceu por mim e que esteve ali, para me ajudar, nas horas difíceis.

À Isabel Dantas(Bebel), por sempre me apoiar, ajudar e por ter fé em mim.

À Livia Lopes Matos, por me ajudar a seguir em frente e não desistir dos meus objetivos durante esses quatro anos de curso.

A todos os meus amigos, principalmente José Lopes da Silva Neto, pelo apoio e torcida.

Aos meus colegas de classe que me ajudaram quando não pude estar presente nas aulas, pela ajuda nos trabalhos, avisos pelo Google Groups etc.

A todos os professores pela paciência com que sempre nos recebeu e pela ajuda nas ausências às aulas que foram necessárias.

A todas as pessoas que me ajudaram direta ou indiretamente neste curso.

RESUMO

Este trabalho apresenta um estudo sobre a tecnologia da TV Digital, o seu surgimento nos países estrangeiros, as possibilidades que se pode obter com sua implantação e as principais mudanças que ela pode desempenhar para os milhões de usuários que possuem e a vêem não só como um meio de comunicação passivo onde o telespectador se comporta como um receptor de informação, mas como meio de muitas outras possibilidades tais como poder responder diretamente as informações passadas pela emissora, escolha individual das informações e usar a TV como meio de comércio eletrônico como já existe na rede mundial de computadores. O estudo mostra uma das características da tecnologia JavaTV, a qual a TV Digital foi desenvolvida e implantada no Brasil, que é a introdução da interação entre emissor e receptor por meio de um canal bidirecional. É abordada a arquitetura de um sistema IDTV, o *Middleware* Ginga onde fica a camada de aplicação da tecnologia e seus subsistemas principais, a interatividade e suas categorias, a plataforma Java e o pacote principal J2ME a qual a tecnologia JavaTV e sua API foram configuradas, os tipos de configurações que existem e qual a utilizada pela TV Digital e como os seus *perfis* são organizados. Por fim, é mostrada uma aplicação por meio de um emulador que simula o ambiente da TV Digital onde o telespectador poderá escolher e desempenhar ações em um programa de televisão interativo finalizando com os resultados obtidos no estudo da tecnologia e na aplicação em questão avaliando o desempenho do emulador utilizado.

Palavras-chave: Middleware. JavaTV. Televisão de alta definição(HDTV). Máquina virtual Java(JVM). Comércio eletrônico.

ABSTRACT

This paper presents a study on the technology of digital TV, its appearance in foreign countries, the possibilities that can be achieved with its implementation and the main changes that she can play for the millions of users who have and see not only as a means liability of communication where the viewer feels like a receiver of information, but as a means of many other possibilities such as to respond directly to the information passed by the station, individual choice of the information and use the TV as a means of e-commerce as exists in the network world of computers. The study shows the characteristics of JavaTV technology, which the Digital TV was developed and implemented in Brazil, which is the introduction of the interaction between sender and receiver through a bidirectional channel. It addressed the architecture of a system IDTV, the Ginga Middleware is the layer where the application of technology and its major subsystems, interactivity and its classes, the Java platform and the main package which the J2ME technology JavaTV and API have been configured, the types of configurations that exist and which is used by Digital TV and how their profiles are organized. Finally, an application is shown by means of an emulator that simulates the environment of the Digital TV where the viewer can choose and perform actions on an interactive television program ends with the results obtained in the study of technology and application in question assessing the performance the emulator used.

Keywords: Middleware. JavaTV. high-definition television (HDTV). Java Virtual Machine (JVM). E-commerce.

LISTA DE FIGURAS

Figura 01 - Bloco Funcional Do Sistema.....	13
Figura 02 - Arquitetura de um sistema de IDTV.....	14
Figura 03 - Arquitetura planejada para o OpenGinga.....	15
Figura 04 - Aplicação EPG utilizando interatividade local.....	19
Figura 05 - Aplicação de e-mail com interatividade intermitente.....	20
Figura 06 - Aplicação de e-mail com interatividade permanente.....	21
Figura 07 - Edições da Plataforma Java 2.....	23
Figura 08 - Configurações Java Micro Edition.....	26
Figura 09 - Estrutura do Profile e da Configuração dentro da J2ME.....	28
Figura 10 - Edições da plataforma java 2 micro edition.....	29
Figura 11 - Arquitetura de um <i>set-top box</i> genérico e um utilizando a tecnologia Java	31
Figura 12 - Ciclo de vida de um Xlet.....	36
Figura 13 - Tela inicial do Software Xletview.....	39
Figura 14 - Aba de aplicações do emulador Xletview.....	40
Tabela 01 – Arquitetura Java TV.....	41
Figura 15 - Emulador Xletview com a aplicação em execução.....	50
Figura 16 - Emulador Xletview com a aplicação em esperando uma ação do usuário.....	55

2.6.2 Profile.....	31
2.7 JAVA TV.....	34
2.7.1 Api java tv.....	35
2.7.2 Pacotes da api java tv.....	36
2.8 XLET.....	38
2.8.1 Xletview.....	40
3 METODOLOGIA.....	42
3.1 SINAL DIGITAL.....	43
3.2 APLICAÇÃO.....	44
3.3 RESULTADOS E DISCURSSÃO.....	45
4 CONCLUSÃO.....	47
5 REFERÊNCIAS BIBLIODGRÁFICAS.....	48

SUMÁRIO

1	INTRODUÇÃO	11
1.1	A POSSIBILIDADE DA INTERATIVIDADE NOS TELEVISORES DOMÉSTICOS.....	13
1.2	JUSTIFICATIVA.....	14
1.3	OBJETIVO GERAL.....	15
1.4	OBJETIVOS ESPECÍFICOS.....	15
2	TV A DIGITAL.....	15
2.1	TV DIGITAL NO BRASIL.....	17
2.2	ARQUITETURA DE UM SISTEMA DE IDTV	19
2.3	MIDDLEWARE GINGA.....	20
2.4	INTERATIVIDADE.....	22
	2.4.1 Interatividade Local.....	23
	2.4.2 Interatividade intermitente.....	24
	2.4.3 Interatividade permanente.....	25
2.5	PLATAFORMA JAVA.....	26
2.6	J2ME.....	28
	2.6.1 Configuração.....	29
	2.6.2 Profile.....	31
2.7	JAVA TV.....	34
	2.7.1 Api java tv.....	35
	2.7.2 Pacotes da api java tv.....	36
2.8	XLET.....	38
	2.8.1 Xletview.....	40
3	METODOLOGIA.....	42
3.1	SINAL DIGITAL.....	43
3.2	APLICAÇÃO.....	44
3.3	RESULTADOS E DISCURSSÃO.....	45
4	CONCLUSÃO.....	47
5	REFERÊNCIAS BIBLIOGRÁFICAS.....	48

INTRODUÇÃO

A televisão é um aparelho de recepção de som e imagem instantânea que capta ondas eletromagnéticas. O primeiro sistema semi-mecânico de televisão analógica surgiu no ano de 1924 em Londres. Há vários tipos de sinais analógicos em todo o mundo e os principais são: *National Television System(s) Committee* (NTSC), *Phase Alternating Line* (PAL), *Séquentiel Couleur avec Mémoire* (SECAM).

O primeiro sinal analógico em cores foi o NTSC. O Brasil utilizou esse sinal por alguns anos até ser substituído pelo o sistema PAL-M, eliminando os vários problemas que o NTSC apresentava.

A primeira transmissão no Brasil aconteceu em 28 de setembro de 1948 na cidade de Juiz de Fora, Minas Gerais, onde foi transmitida uma partida de futebol. E a primeira transmissão em cores aconteceu na festa da uva em 19 de fevereiro de 1972 na cidade de Caxias do Sul. A partir destes acontecimentos, a TV analógica evoluiu em qualidade, quantidade e desde sua criação, é sinônimo de modernização e sucesso. Com a chegada do novo sistema de transmissão digital no padrão Japonês, o Serviço Integrado de Transmissão Digital Terrestre (ISDB-T), o telespectador não só recebe informações como também pode interagir com a TV.

A TV digital é uma realidade muito próxima e que promete revolucionar o conceito de assistir TV, já que o número de televisores no Brasil atinge números grandiosos e que indiscutivelmente é um sucesso da tecnologia mundial. Com a chegada do sinal digital, as informações pela televisão poderão ter um novo comportamento. Além do melhor sinal, melhorando o som e a imagem, a TV digital poderá ter a transmissão de vários programas em um mesmo canal, possibilitando, os telespectadores escolherem o que assistir. Deste modo, o grande diferencial que a TV digital pode oferecer são os serviços interativos. Combinada com um canal de retorno, o telespectador poderá usufruir do conforto de fazer compras na sua própria casa e também selecionar programas sobre demanda, tudo isso pelo controle remoto da sua televisão.

No mundo, os três mais importantes *middleware* para TV Digital são: o europeu (MHP do DVB), o japonês (ARIB do ISDB) e o americano, o DASE do ATSC. Estes três middlewares tem em comum o seguinte ponto: o uso da linguagem de programação Java que é uma unanimidade e todos usam a biblioteca Java TV como base para suas aplicações.

Considerando estes pontos, torna-se evidente que a utilização no mercado mundial é promissor e que vale a pena aprofundar as pesquisas nesta plataforma.

O sinal é emitido via satélite, via cabo ou broadcast terrestre. A recepção do sinal é feita através dos aparelhos de TV, seguindo o modelo de funcionamento do sinal analógico, e juntamente com ele ou dentro dele, um decodificador para sinal digital (set-top-box). Esse decodificador, inicialmente será vendido separadamente para que os televisores possam receber o sinal, e se adaptar a esta nova realidade.

Nacionalmente, o governo criou o Comitê do Sistema Brasileiro de Televisão Digital (SBTVD) visando realizar estudos técnico-econômicos de viabilidade para essa nova tecnologia. Em 23/06/06 o governo brasileiro comunicou oficialmente a escolha do padrão japonês de TV digital, e a definição ocorreu através de decreto assinado em 29/06/06. **(Decreto 5.820).**

Depois do decreto, foi criada a camada de aplicação chama de Ginga-NCL, desenvolvida pela PUC - Rio e pela Universidade Federal da Paraíba. A camada de aplicação é uma parte do software que faz a ligação entre o hardware dos *Set-top-box* e as aplicações da TV interativa. A PUC foi responsável pelo ambiente Ginga (*Ginga-Nested Context Language* (NCL) que trata o sincronismo das mídias e adaptabilidade dos múltiplos dispositivos de exibição. Esta forma tem como vantagem o pequeno tamanho de suas aplicações facilitando a transmissão da TV digital. A UFPB foi responsável pelo ambiente procedural Ginga-J, extensão do *Globally Executable MHP* (GEM), que determina uma lista de pacotes que permitem a execução e acesso de determinadas tarefas que serão usadas para que o Ginga-J possa funcionar. As principais Programações de Aplicação da Interface (API's) usadas para o desenvolvimento de aplicações para a TV digital são a java TV da SUN e a Microsoft TV. No padrão digital Video Broadcasting (DVB), existe o *Multimedia Home Platform* (MHP) que define as tecnologias para interação digital e visa à adaptação da Internet, proporcionando assim, interatividade digital no terminal do usuário (ABNT, 2008).

A plataforma Java pode se adaptar a vários tipos de aplicações como, por exemplo: jogos em grupo, conexão de pessoas do mundo inteiro, calculo da taxa de juros de um financiamento, aplicativos de intranet e de outras soluções de comércio eletrônico que formam a base da computação corporativa. Até hoje, a plataforma Java já atraiu mais de 6,5 milhões de desenvolvedores de software. É utilizada em todos os principais segmentos da indústria, estando presente em uma ampla gama de dispositivos, computadores e redes computacionais.

A linguagem Java foi pensada originalmente para se utilizar em qualquer tipo de eletrodoméstico, mas a idéia fracassou. Um dos fundadores da sun resgatou a idéia para utilizá-la no âmbito da Internet e converteram o Java em uma linguagem potente, segura e universal por ser gratuita e utilizada por todos. Um dos primeiros triunfos do Java foi à integração no navegador Netscape e a possibilidade de executar programas dentro de uma página web, até então impensável com o HTML.

Atualmente java é utilizado em um amplo leque de possibilidades. Praticamente qualquer outro trabalho programado em uma linguagem diferente, pode ser efetuado também em Java e muitas vezes com grandes vantagens. Com Java, podem ser programadas páginas web dinâmicas com acesso à base de dados utilizando XML com qualquer tipo de conexão de rede entre qualquer sistema. Em geral, qualquer aplicação que desejarmos fazer com acesso através web se pode fazer utilizando Java (SUN MICROSYSTEMS, 2008).

Sua versatilidade, eficiência, portabilidade de plataforma e segurança fazem dela a tecnologia ideal para a computação em rede. A tecnologia Java está em todo lugar, ela pode ser encontrada em laptops, consoles de jogo, supercomputadores científicos, telefones celulares e até nos decodificadores utilizados para receber o sinal digital onde o usuário doméstico pode acessar informações relativas ao programa que está assistindo, consultar saldo bancário, alugar vídeos, se divertir com jogos 3D e muito mais. Desta forma, este usuário poderá usufruir, de forma completa e interativa, das melhorias trazidas pela TV digital. Por esses motivos que a linguagem Java foi à escolhida para ser a linguagem padrão da TV digital e também dessa pesquisa.

Essa pesquisa está organizada do seguinte modo: o capítulo 1, justificativa do assunto, objetivos gerais e específicos. O capítulo 2 trata da fundamentação teórica envolvida na pesquisa, da TV digital e suas tecnologias. O capítulo 3 mostra toda a metodologia. Por fim, no capítulo 4 serão descritos os resultados e conclusão dessa pesquisa.

1.1 A POSSIBILIDADE DA INTERATIVIDADE NOS TELEVISORES DOMÉSTICOS

Uma das principais características da TV digital é sem dúvida, a interatividade que o telespectador pode ter com essa nova tecnologia. Através de um canal de retorno que existe no set-top-box dos aparelhos domésticos, os mesmos poderão enviar informações, no sentido receptor emissora. Desse modo, Essas mesmas informações são respondidas pela emissora

para o telespectador. Partindo desta situação, de que forma a tecnologia java pode ser utilizada para o desenvolvimento da interatividade da TV digital?

1.2 JUSTIFICATIVA

O tema foi escolhido devido à grande importância dessa tecnologia que já faz parte da cultura do brasileiro. O novo sinal digital pode fazer com que essa cultura só aumente e dê oportunidades a outros ramos da indústria e comércio. Uma vez que, o telespectador não tem o poder de escolha e nem de realizar ações de compra ou de comunicação como acontece na internet. Essa tecnologia se compara, no aspecto de grandiosidade, a mudança da TV preto e branco para TV colorida como aconteceu na metade do século passado.

A cultura de assistir televisão é intensa, não só no Brasil como no mundo inteiro. É um negócio muito lucrativo para a economia de vários países, desde os mais variados setores da indústria e comércio, lançamentos de novos produtos, como também as tendências de moda, estilo de vida, pois as notícias no mundo inteiro que ela pode mostrar em seus programas e propagandas elevam a grandes audiências e conseqüentemente geram empregos nas diversas áreas, desde técnicos a apresentadores de TV.

Com a TV digital, o telespectador poderá interagir com a TV de diversas formas como também poderá ser usada não só como aparelho de televisão em si, mas em outros equipamentos como: celular, computadores de mão, relógios e qualquer equipamento que possua uma tela e seja receptora do sinal digital. A TV digital oferece algumas novidades que merecem ser citadas como, por exemplo, a possibilidade de interação que é a capacidade do telespectador poder enviar informação a TV, através do controle remoto, e não apenas receber, capacidade essa que já existe em outros países. O usuário da TV digital pode fazer negócios usando o controle remoto da televisão enquanto assiste a ela, como por exemplo, comprar uma camisa de futebol enquanto vê seu time jogar, como também ver informações do time, jogadores, histórico de cada um deles e etc.

Esse tipo de negócio é chamado de comércio na televisão (t-commerce) é mais um passo do comércio eletrônico (e-commerce) onde se pode comprar usando o mouse na internet. Esses dois ramos do e-commerce podem existir sem prejudicar um ao outro já que as empresas poderão fazer com que os consumidores comprem, também, usando o controle remoto.

O telespectador que não possui Internet poderá usufruir de muitos aspectos que a internet tem a oferecer na sua TV digital. Esse é um ramo que não pode ser ignorado. Muitos conteúdos digitais como entretenimento, negócios e as diversas áreas podem gerar demandas para os desenvolvedores suprir essa tendência de mercado.

1.3 OBJETIVO GERAL

Compreender os conceitos e ferramentas orientadas ao desenvolvimento da TV digital bem como desenvolver uma aplicação usando a API Java TV para que o usuário interaja com ela recebendo e enviando informações.

1.3.1 Objetivos Específicos

- Obter conhecimento na tecnologia Java TV, conhecendo suas API's, observando seu funcionamento e características;
- Estudar o emulador xletview, como se dá o seu uso, como criar uma xlet e observar suas vantagens e desvantagens;
- Dominar o conteúdo do Framework Ginga-J, funcionamento e capacidades;
- Criar uma aplicação em Java TV no emulador xletview que possa ser capaz de mostrar a interatividade para o telespectador.

2 TV DIGITAL

A partir da década de 30, o mundo passou a conhecer um aparelho chamado de televisão, e desde então, este se tornou um grande veículo de comunicação gerando grande impacto em diversas áreas do conhecimento como a educação, cultura, esporte, comércio, indústria, política e entretenimento.

A televisão é resultado do desenvolvimento científico e tecnológico de diversos pesquisadores. “É importante ressaltar que, em nenhum momento, um estágio evolutivo da televisão substituiu o estágio anterior; a evolução sempre foi lenta e gradual, se agregando paulatinamente ao modelo anterior” (Becker et alli, 2004).

No final da década de 80, os japoneses começaram a estudar o conceito para desenvolver um novo serviço de tecnologia digital e apresentaram o *High-definition television* (HDTV) que é um sistema com maior qualidade de imagem e som comparados aos de cinema, que proporciona maior realismo.

Mas, somente nos anos 90, é que a tecnologia de televisão digital deu um passo bastante grande com o surgimento de três padrões mundiais de *middleware*: em setembro de 1993, o padrão de televisão digital europeu *Digital Video Broadcasting* (DVB), surgiu devido ao resultado do trabalho de indústrias do ramo televisivo, emissoras, fabricantes, operadoras de rede, desenvolvedores de software e órgãos regulamentadores (DVB, 2005).

O segundo padrão é o americano *Advanced Television Systems committee* (ATSC). Lançado em 1998 pela ATSC, como uma organização internacional sem fins lucrativos para o desenvolvimento de padrões de televisão digital. Este comitê é formado por várias organizações e vários países já adotaram esse padrão, tais como Argentina, México, Canadá e Coreia do Sul.

Em 1999, o grupo *Digital Broadcasting Experts Group*, conhecida também como DiBEG, apresentou o padrão japonês de televisão digital denominado de *Integrated Services Digital Broadcasting* (ISDB).

Um conjunto de tecnologias foi criado ou adaptado para viabilizar a execução de aplicações que permitem interatividade através da televisão, surgindo então, os sistemas de televisão digital que em linhas gerais, definem um grupo de componentes que atuam em conjunto para viabilizar a transmissão, recepção e apresentação dos conteúdos audiovisuais e a execução dos programas interativos em aparelhos de televisão. Os sistemas são baseados em padrões, que regem desde a codificação do vídeo até a camada de abstração do software permitindo que aplicações sejam executadas em diferentes equipamentos (Batista, 2006).

Com a combinação de novos e eficientes esquemas de modulação para transmissões digitais e com a evolução dos algoritmos de codificação de vídeo, surge a TV Digital, como uma revolução tecnológica de grande alcance, já que grande parte da população mundial passa parte de seu dia na frente de um aparelho televisor. O telespectador passará a ter um papel ativo nas programações que antes ele apenas assistia. O controle remoto deixará de fazer as funções simplesmente operacionais e possibilitará ao telespectador, escolher o modo

de como ver um programa intervindo diretamente no mesmo, como por exemplo, escolher o ângulo da câmera para assistir um jogo de futebol com também, fazer compras no meu do programa. As possibilidades serão grandes, as características do computador transportadas para dentro do ambiente da televisão.

2.1 TV DIGITAL NO BRASIL

No Brasil, o padrão adotado foi o japonês e o funcionamento da TV digital se deu em 02/12/2007, porém, apenas algumas capitais tem esse serviço disponível. O comitê do sistema brasileiro de televisão digital, afirma que essa migração será lenta, mas com avanços constantes em relação ao território, começando pelas capitais e depois a todo o território nacional.

Segundo o professor Carlos Ferraz, diretor-adjunto do Centro de Estudos e Sistemas Avançados do Recife (Cesar), essa nova tecnologia não deve substituir o computador pessoal. O sistema brasileiro usa o *Moving Picture Experts Group* (MPEG-4) para codificação e MPEG-2 para transmissão. Apesar de ter sido alvo de muitas críticas, esta escolha requer menos banda de transmissão, pois possui uma capacidade de processamento maior, configurando uma excelente vantagem sobre os demais métodos.

A TV digital abre um vasto leque de possibilidades, convergindo às tecnologias presentes no nosso cotidiano e colocando o telespectador como membro ativo da cadeia televisiva.

A TV digital traz várias novidades quando comparada à TV analógica, como por exemplo:

- Som com a qualidade de *Compact Disk* (CD);
- Imagem com a qualidade de *Digital Video Disk* (DVD) *player* com 525 linhas;
- Incremento nas transmissões de áudio associadas aos programas, como a dublagem e a legenda em vários idiomas
- Transmissão de dados associados aos programas;
- Otimização da área de cobertura, possibilitando uma maior flexibilidade para ajustar os parâmetros de transmissão de acordo com as características geográficas locais;
- Serviços adicionais de telecomunicações (comércio eletrônico);
- Maior número de canais;

- Maior diversidade de programação;
- *Single Frequency Network* (SFN), uma tecnologia capaz de definir, por exemplo, um único número de canal para todo o Brasil;
- Funcionalidades disponíveis em um *Personal Video Recorder* (PVR). Este dispositivo permite, principalmente, a gravação digital em alta performance dos conteúdos multimídia preferidos pelo usuário, como programas, filmes, imagens e fotos;
- Nova gama de aplicações e serviços interativos englobando texto, vídeo, áudio e os mais variados tipos de elementos gráficos.
- A possibilidade de o telespectador escolher o que deseja ver, como, quando e onde (DVB SERVICES, 2005).

O sistema brasileiro foi projetado com uma boa proteção contra ondas de rádio proporcionando uma qualidade superior de som e imagem para os telespectadores de TV. O sistema é composto por sete blocos funcionais, o SBTVD – 01 ao SBTVD – 07. Estes são os responsáveis pela codificação de canais, modulação, vídeo, áudio, e também pelos procedimentos de multiplexação, descrição do terminal de acesso (Set-top-box), propriedade de direitos intelectuais. O diferencial principal da estrutura de transmissão deste sistema é o SBTVD – 06 e 07, que são responsáveis pela descrição do uso do *middleware* e interatividade, padronizando o canal de retorno das aplicações de interatividade. A figura 1 mostra como é organizado o bloco funcional do sistema (ABNT, 2008).

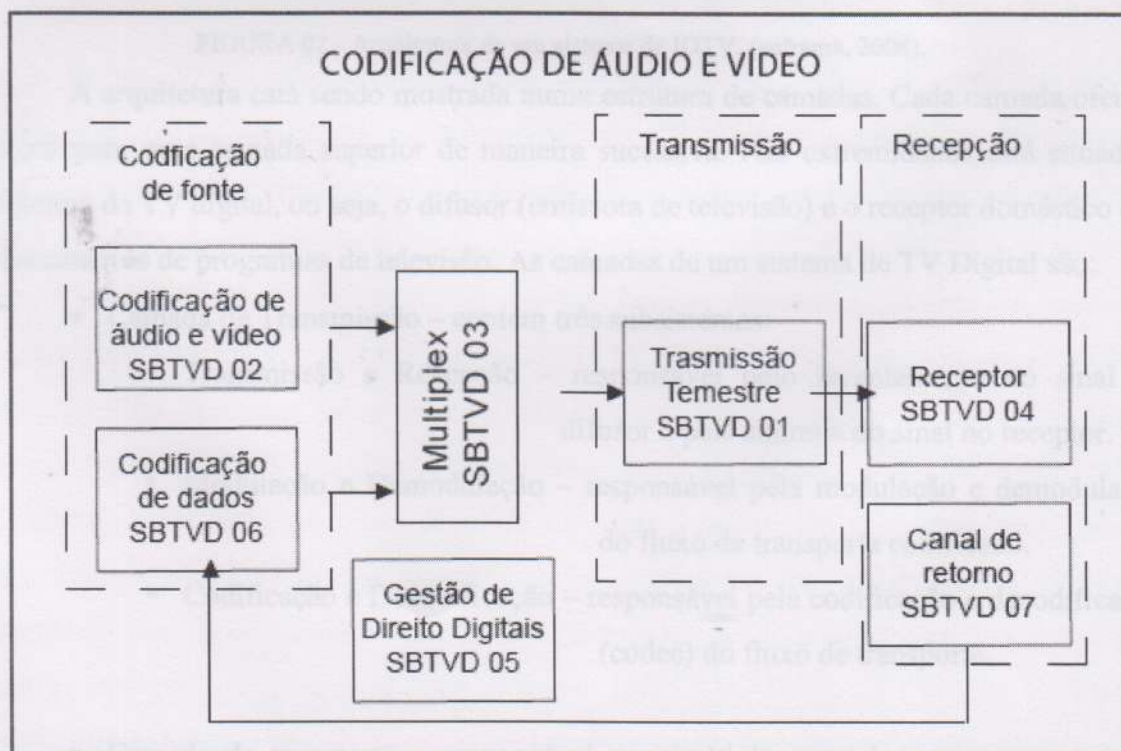


FIGURA 01: BLOCO FUNCIONAL DO SISTEMA (ABNT, 2008).

2.2 ARQUITETURA DE UM SISTEMA DE IDTV

A figura 2 apresenta a arquitetura de um sistema de TV digital interativa. Uma arquitetura corretamente representada pode assegurar que um sistema irá satisfazer uma série de requisitos relacionados à performance, confiabilidade, portabilidade, escalabilidade e interoperabilidade (Garlan, 1996), mostrando os principais componentes, incluindo suas interações e omitindo os detalhes que não são pertinentes as interações entre os componentes.



FIGURA 02 - Arquitetura de um sistema de IDTV. (suframa, 2008).

A arquitetura está sendo mostrada numa estrutura de camadas. Cada camada oferece serviços para uma camada superior de maneira sucessiva. Nas extremidades está situada a arquitetura da TV digital, ou seja, o difusor (emissora de televisão) e o receptor doméstico que são os usuários de programas de televisão. As camadas de um sistema de TV Digital são:

- Camada de Transmissão – contam três subsistemas:
 - Transmissão e Recepção – responsável pelo levantamento do sinal no difusor e pela sintonia do sinal no receptor.
 - Modulação e Demodulação – responsável pela modulação e demodulação do fluxo de transporte codificado.
 - Codificação e Decodificação – responsável pela codificação e decodificação (codec) do fluxo de transporte.
- Camada de transporte – responsável por multiplexar vários programas em um único fluxo de transporte no ambiente da emissora. No

ambiente do usuário, realiza a demultiplexação do fluxo de transporte de acordo com o que o usuário selecionou.

- Camada de Compressão – é responsável por comprimir sinais de áudio e vídeo no ambiente da emissora (difusor) e descomprimir esses sinais no ambiente do telespectador.
- Camada de *Middleware* - é a camada de software que oferece o serviço para a camada de aplicação, escondendo as peculiaridades e heterogeneidade das camadas de compressão, transporte e transmissão. O estudo dessa monografia se baseia no *middleware* Ginga que será visto no próximo capítulo.
- Camada de Aplicativos – corresponde pelo que o usuário vai poder visualizar e interagir suportadas pelas camadas inferiores. Responsável pela execução dos filmes, Internet, etc.

2.3 MIDDLEWARE GINGA

Ginga é a camada de software intermediário (*middleware*) que permite o desenvolvimento de aplicações interativas para a TV digital de forma independente da plataforma de hardware dos fabricantes de terminais de acesso (set-top boxes). Foi o primeiro *middleware* opensource desenvolvido no Brasil lançado em 03 de Julho de 2007 no auditório militar de engenharia Praia Vermelha, Rio de Janeiro, Resultado de anos de pesquisas lideradas pela Pontifícia Universidade Católica(PUC-Rio) e pela Universidade Federal da Paraíba (UFPB). O Ginga reúne um conjunto de tecnologias e inovações brasileiras que o tornam a especificação de *middleware* mais avançada e, ao mesmo tempo, mais adequada à realidade do país. Este pode ser dividido em dois subsistemas principais permitindo o desenvolvimento de aplicações seguindo dois paradigmas de programação diferentes. Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma possuirá uma melhor adequação que o outro (PORTAL DO SOFTWARE PÚBLICO BRASILEIRO, 2008).

O *middleware* ginga pode ser dividido em três subsistemas principais: Ginga-CC, Ginga-J e Ginga-NCL.

- Ginga-CC (*Ginga Common-Core*) oferece o suporte básico para os ambientes declarativo (Ginga-NCL) e procedural (Ginga-J). Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma de programação (declarativo ou procedural) possuirá uma melhor adequação que o outro.
- Ginga-J foi desenvolvido pela UFPB para prover uma infra-estrutura de execução de aplicações baseadas em linguagem Java, com facilidades especificamente voltadas para o ambiente de TV digital. A arquitetura Ginga-J é composta de cinco camadas: Hardware, Sistema Operacional, Implementação Ginga e Java Virtual Machine (JVM), API Ginga-J e *Xlets*. Merece destaque a camada de aplicação onde são implementadas as *xlets* que são aplicações para a TV Digital, similares as Applets em Java, mas não são iguais. A *SUN Microsystems* desenvolveu, também, um emulador para desktop onde as *xlets* são executadas chamdado de *Xletview*.
- Ginga-NCL (*Nested Context Language*) O subsistema lógico NCL é composto por uma série de módulos e o mais importante é o NCL Formatter que controla o recebimento e a sua apresentação, organizando o relacionamento entre objetos de mídia. Com a linguagem NCL é feita descrição dos objetos e partir daí eles são sincronizados para que o espaço temporal deles seja adequado, foi desenvolvido pela PUC-Rio para prover uma infra-estrutura de apresentação de aplicações baseadas em documentos hipermídia escritos em linguagem NCL, com facilidades para a especificação de aspectos de interatividade, sincronismo espaço-temporal de objetos de mídia, adaptabilidade e suporte a múltiplos dispositivos. NCL possui a linguagem de programação Lua como sua linguagem de script (PORTAL DO SOFTWARE PÚBLICO BRASILEIRO, 2008).

Para facilitar o desenvolvimento de aplicações Ginga-NCL, a PUC-Rio desenvolveu a ferramenta Composer, ambiente voltado para programas NCL para TV Digital interativa. Nessa ferramenta a abstração é definida em diversos tipos de visões que simula um tipo específico de edição, exemplo: textual, estrutural, temporal e leiaute. Essas visões funcionam de maneira sincronizada para possibilitar um ambiente integrado de autoria (ABNT, 2008).

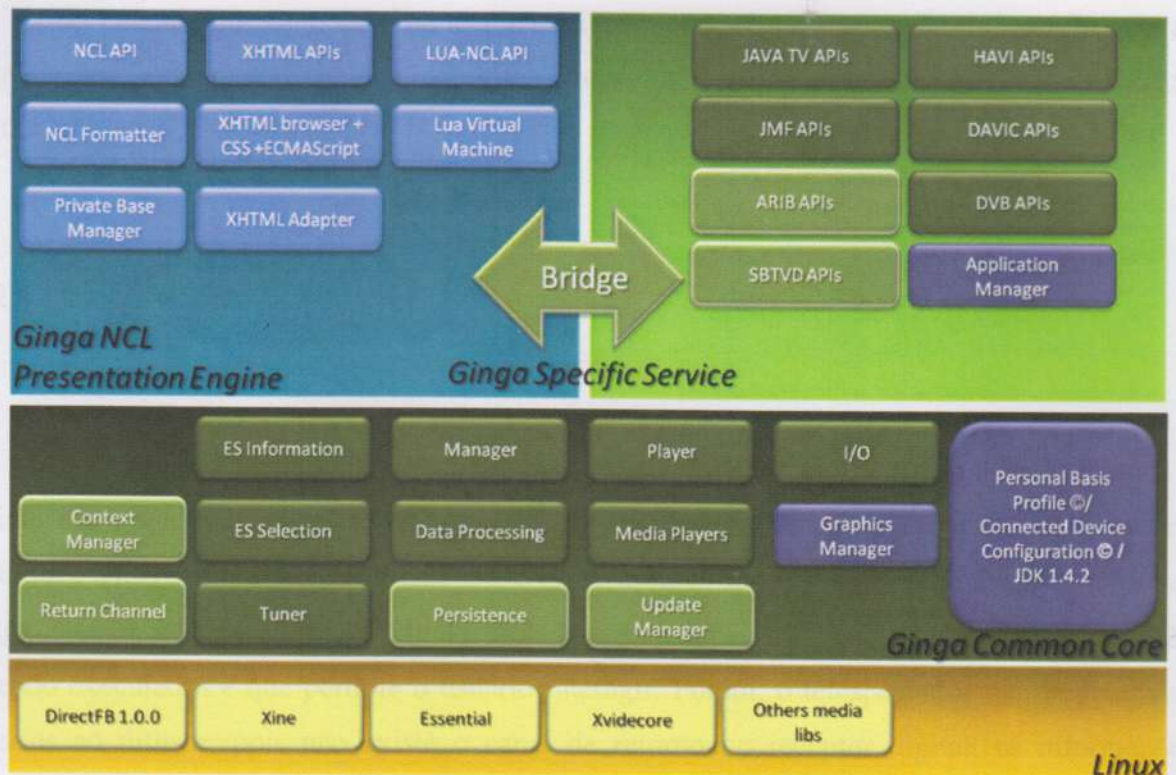


FIGURA 03: arquitetura do Ginga (Ginga DF, 2008).

2.4 INTERATIVIDADE

A evolução tecnológica tem se desenvolvido bastante nos últimos anos na área da telecomunicação e na rede mundial de computadores. Hoje, com a internet o usuário pode interagir com programas, conteúdos e serviços. Com isso, os usuários criaram uma expectativa de poderem ter essa interação, na televisão, como acontece na Internet.

Cria-se uma nova demanda no mercado mundial para o desenvolvimento de aplicações para a televisão por meio do controle remoto ou por meio de um teclado. Fica claro que o diálogo pode existir entre o telespectador e os programas, conteúdos e serviços que a Internet oferece.

Através dessa nova tecnologia, as emissoras podem dispor de ferramentas para aumentar a audiência e conseqüentemente, os lucros com anúncios, o aumento do tempo na frente de uma televisão por parte dos telespectadores. Porém, o sucesso da tecnologia digital para a televisão vai depender do modelo de negócio adotado por cada emissora para inovar na elaboração dos novos serviços que a interatividade pode proporcionar e com isso atrair ainda mais os usuários para o consumo desses dos mesmos.

Falando do ponto de vista técnico, há três categorias de interação do usuário com as aplicações, serviços e conteúdos interativos: local, intermitente e permanente (FERNANDES, 2004).

2.4.1 Interatividade Local

A interatividade local é a mais básica das três categorias. O serviço de difusão gera o sinal dos programas de televisão para que o canal de difusão possa transmitir os fluxos de áudio e vídeos, unidirecionalmente para os receptores domésticos. A antena recebe esses fluxos enquanto que o receptor digital fica responsável pelas aplicações interativas que estão sendo executadas e que permite o usuário interagir. Porém, o telespectador não pode enviar dados ao difusor, pois não existe o canal de retorno no receptor digital, as informações enviadas pelo difusor são de caráter geral para todos os usuários caracterizando-os como meros telespectadores. Para dar um exemplo para esse tipo de interação, pode-se citar a configuração de legendas, jogos residentes, guias de programação, entre outros.

A figura 4 mostra um guia de programação que informa ao usuário a programação precisa de cada emissora associada ao gênero (comédia, drama, suspense, documentário, etc.), horário, duração, resumo de cada programa, dentre outras informações.

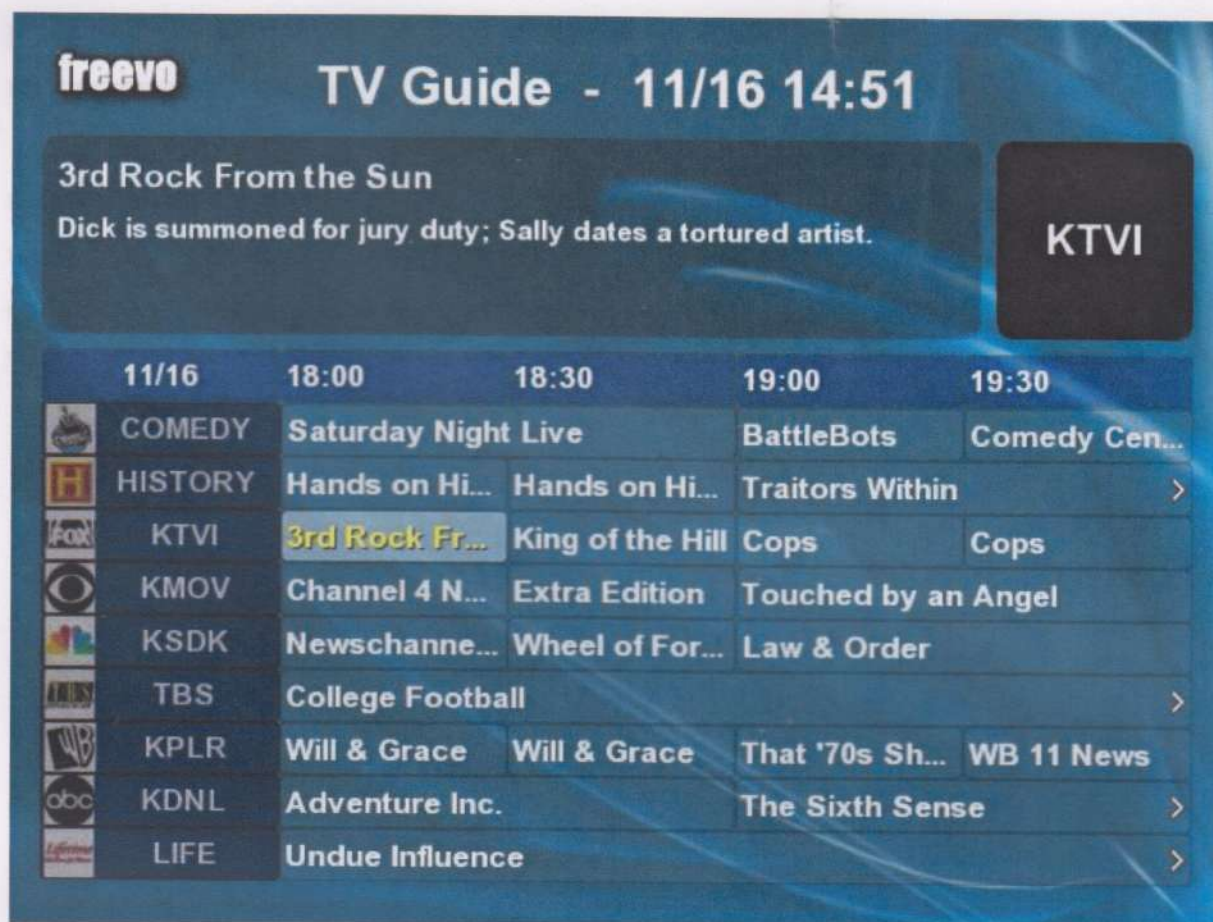


FIGURA 04 - Aplicação EPG utilizando interatividade local. (Freevo, 2005).

2.4.2 Interatividade Intermitente

Na interatividade intermitente há algumas mudanças em relação à local, uma delas é que além do provedor de serviço de difusão, há também o provedor de serviço de interação, onde vai ser possível a comunicação do usuário em direção ao difusor.

O provedor de serviço de difusão continua fazendo as mesmas atividades que são de gerar o sinal dos programas para a o canal de difusão unidirecional onde é transmitido o vídeo e o áudio ao receptor doméstico e a antena doméstica também continua a receber os fluxos de áudio e vídeo. Já no receptor digital, as mudanças são significativas, o canal de retorno que não existia na categoria Local agora está presente no receptor que poderá transmitir os fluxos de dados ao difusor que tratará esses fluxos no provedor de serviço de interação. Porém, o provedor de serviço de interação que existe nessa categoria é unidirecional de forma que não é

possível que o difusor retorne a informação ao telespectador, caracterizando esse canal de retorno como não-dedicado.

Para esse tipo de interação, com um canal não-dedicado, são geralmente usadas aplicações como votações, enquetes e pesquisas de opinião onde o telespectador não espera uma resposta do difusor pelo canal de retorno. A figura 5 mostra uma tela de aplicação SMS, onde o telespectador pode enviar uma mensagem ao difusor através do controle remoto.



FIGURA 05 - Aplicação SMS utilizando interatividade intermitente. (Jucá, 2005).

2.4.3 Interatividade Permanente

A interatividade permanente é uma evolução bastante significativa, nela, o provedor de serviço de interação funciona com um canal bidirecional, diferente da categoria interatividade intermitente que usa um canal unidirecional. Assim, os receptores digitais

domésticos recebem o sinal bidirecional, do difusor, onde os fluxos de dados são transmitidos para o receptor.

Com essa categoria de interação usando um canal de retorno dedicado, é possível ter acesso as funções básicas de um computador conectado a Internet, enviando e recebendo e-mails, chats, navegação, jogos multiusuários em tempo real, compras online, *homebanking*, educação a distância, etc. A outra novidade é a comunicação entre telespectadores e não só entre telespectador e o difusor.

A figura 6 mostra uma aplicação onde o telespectador pode enviar, receber, apagar e armazenar e-mails através de um controle remoto apropriado ou por um teclado.



FIGURA. 06 - Aplicação de e-mail com interatividade permanente. (Davenport, 2005).

2.5 PLATAFORMA JAVA

A plataforma Java veio para ficar, a mais de 10 anos no mercado e ainda na liderança entre outras linguagens, segundo a pesquisa *Tiobe Programming Community Index*, que

classifica a popularidade de linguagens de programação, Java vem em primeiro lugar como no ano passado, em seguida vêm C, Visual Basic, PHP, C++, Perl, Python e C#. Mais abaixo na lista estão Delphi, JavaScript e Ruby. Os números da Tiobe se baseiam na disponibilidade mundial de engenheiros qualificados, cursos e fornecedores e na utilização de ferramentas de busca populares para calculá-los. A Linguagem Java - Java Virtual Machine (JVM), API e bibliotecas – são indiscutivelmente muito boas e grandiosamente populares. No Brasil, a demanda por pessoal qualificado cresce cada vez mais e a procura por capacitação na linguagem Java também, pois, a expansão do mercado, principalmente dos dispositivos móveis, tem aumentado (TIOBE, 2008).

Para testar as aplicações, que se chamam xlets, de uma maneira fácil e rápida dentro do ambiente da TV digital em desktops ou notebooks, usa-se um emulador chamado de xletView baseado na *middleware* MHP que é Open Source (SUN MICROSYSTEMS, 2008).

Uma nova estratégia da Sun é abraçar novas tecnologias adotando outras que já existem fazendo com que a plataforma se torne forte em todos os domínios.

A API apresenta uma abstração de alto nível, possui uma característica que facilita no desenvolvimento das aplicações, pois o desenvolvedor não se preocupa com os protocolos de serviços, transmissão e rede. Funciona como um *middleware*, porque se situa entre o sistema operacional e as aplicações, o *middleware* é um neologismo onde se concentra as camadas de um software, é nele que os serviços de identificação, autenticação, autorização, diretórios, certificados digitais e outras ferramentas de segurança, estão aglutinados, pois, nas aplicações modernas, esses serviços são delegados pela cada de *middleware* (MACIEL, 2004).

Para mostrar os conteúdos dessa nova tecnologia, é preciso que a TV seja moldada de forma que seus conteúdos possam ser criados, e para isso, são usados os *middlewares*, são eles que descrevem os formatos de transmissão e codificação, são camadas intermediárias entre o hardware e o sistema operacional, seu objetivo é esconder os detalhes da plataforma de hardware e sistema operacional para que as aplicações sejam mais simples.

Há três tipos diferentes de pacotes Java:

- *Java 2 Standard Edition* (J2SE): versão básica para desktop e estações de trabalho, essa versão é possível desenvolver a maior parte das aplicações.

- *Java 2 Enterprise Edition* (J2EE): para aplicações de grande porte, nesse pacote o uso do enterprise java beans (EJB) é muito grande como também de servidores de aplicação e servlets, a implementação padrão chama-se Java 2 software development kit enterprise edition (J2SDKEE).

- *Java 2 Micro Edition (J2ME)*: destinado para desenvolvimento de aplicações em dispositivos móveis ou de pequeno porte (SUN MICROSYSTEMS, 2008). Na figura 7, são mostrados os três tipos de pacotes Java:

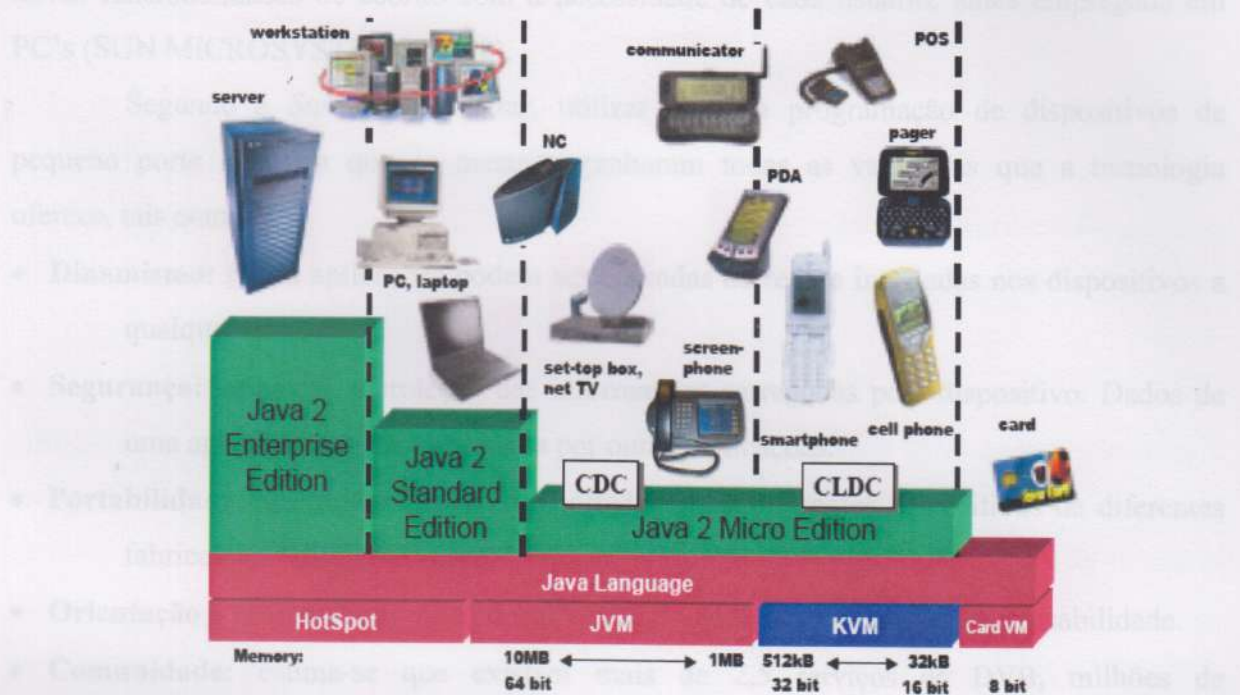


FIGURA 07 - EDIÇÕES DA PLATAFORMA JAVA 2 (SUN MICROSYSTEMS, 2008).

2.6 J2ME

No mundo todo, a aplicação para dispositivos móveis tem crescido cada vez mais, as grandes empresas de software e outras tecnologias têm prestado muita atenção nessa área já que o crescimento dessa tecnologia tem alcançado números promissores.

Junto com as outras edições da plataforma Java, a edição micro do Java (J2ME), tem sido de suma importância para as empresas de tecnologia pesquisadoras e executoras de programas, que fazem sucesso no mercado e usam como base as outras edições java como a *Enterprise Edition* e a *Standard Edition*.

A evolução dos computadores, desktops, passando para a era móvel, não só os PC's como também os aparelhos celulares e tantos outros dispositivos móveis, surgem para dar comodidade e versatilidade para os consumidores que mais precisam de informação na palma

da mão, a J2ME veio para ficar e é nessa edição que a TV digital vai receber as aplicações necessárias para que se torne interativa.

A J2ME é um pacote java voltado para aplicações em dispositivos móveis utilizada para dar toda estrutura tecnológica, dando possibilidade de desenvolver, atualizar, e instalar novas funcionalidades de acordo com a necessidade de cada usuário, antes empregada em PC's (SUN MICROSYSTEMS, 2008).

Segundo a *Sun Microsystems*, utilizar java na programação de dispositivos de pequeno porte significa que os mesmos ganharam todas as vantagens que a tecnologia oferece, tais como:

- **Dinamismo:** novas aplicações podem ser baixadas da rede e instaladas nos dispositivos a qualquer tempo.
- **Segurança:** garantem a proteção das informações carregadas pelo dispositivo. Dados de uma aplicação não são acessados por outras aplicações.
- **Portabilidade:** aplicações podem ser portadas entre diferentes dispositivos de diferentes fabricantes e diferentes tipos.
- **Orientação a Objetos:** alto nível de abstração do código, modularização, reusabilidade.
- **Comunidade:** estima-se que existam mais de 2,5 serviços de DVB, milhões de desenvolvedores no mundo e Java vem se tornando a linguagem de programação mais popular ensinada em cursos de atualização e universidades.
- **Confiabilidade:** não é tolerado reinicializações ou quedas em dispositivos embargados. Graças aos mecanismos de proteção e gerência de memória, Java atinge essa demanda.

2.6.1 Configuração

A configuração define o menor ponto em comum dos recursos da plataforma java suportado por todos os dispositivos com características iguais que pertencem a uma determinada categoria, tanto pela própria linguagem *Java* como pela máquina virtual, assim como suas bibliotecas de classes e APIs (SUN MICROSYSTEMS, 2008).

A tecnologia J2ME foi dividida em alguns grupos, cada um, voltado ao porte de dispositivos com o objetivo de resolver o problema do subaproveitamento de recursos, a esta divisão, dar-se o nome de configuração.

Atualmente há dois tipos de configuração existente na plataforma J2ME. São elas:

- CDC – *Connected Device Configuration*: trata de dispositivos de memória e processamento com mais poder do que CLDC, destinado para dispositivos com uma conexão permanente de rede e um mínimo de 2 a 16 MB de memória disponível para o sistema Java, podem ser ligados por fio. Essa configuração é usada pelo Set-top-boxes para TV digital.
- CLDC – *Connected Limited Device Configuration*: entre 160 a 512 KB de RAM disponível para o sistema Java e uma intermitente (limitada) conexão de rede, processador de 16 ou de 32 bits. Dispositivos CLDC são normalmente *wireless* com largura de banda limitada (normalmente 9600 bits ou menos), *paggers*, PDAs, etc. *Kilo Virtual Machine* (KVM).

O CDC é um conjunto de tecnologias que tem a capacidade de construir interfaces gráficas mais poderosas e aplicações mais inteligentes e pesadas em termos computacionais e é utilizada em dispositivos mais potentes. Foi qualificada para dispositivos fixos um pouco maiores, em relação à configuração CLDC, com pelo menos alguns megabytes de memória disponível e alguma forma de acesso a Internet usando a Máquina Virtual Clássica, com isso, possibilidade de implementação completa de sua especificação (SUN MICROSYSTEMS, 2008).

O CLDC é um conjunto de API's específicas para ambientes extremamente restritos, com a capacidade mínima de processamento, fonte de energia limitada, display reduzido, conectividade intermitente etc. É nesse ambiente de execução que o CLDC trabalha, voltado para dispositivos de pequeno porte, como o celular, comparado com o CDC. Os objetivos da CLDC são:

- Dar suporte a uma grande variedade de dispositivos pequenos, desde dispositivos de comunicação, como aparelhos celulares aos *personal organizers*, terminais de venda entre outros.
- Para não ser preciso ter a compreensão completa sobre o sistema nativo disponível nos dispositivos, com isso um conhecimento mínimo é suficiente.
- Definir um o “menor denominador comum” da tecnologia *Java* aplicada a essa grande variedade de dispositivos móveis.

A figura 8 mostra as configurações CDC e CLDC.

os dispositivos a um mesmo segmento vertical de mercado, onde as aplicações não percam sua funcionalidade com a mudança de um dispositivo de mesma configuração.

A plataforma java está em plena e intensa evolução onde são criados novos perfis definidos através da *Java Community Process*. Uma particularidade interessante que deve ser mencionada é a capacidade de dentro os perfis de maior capacidade, manter a compatibilidade com os perfis mais restritos.

Os perfis existentes hoje dentro da configuração CDC são três, o *Foundation Profile*, o *Personal Basic Profile* e o *Personal Profile*. O primeiro é o mais básico e não dá suporte a *Graphical User Interface* (GUI). O segundo é mediano e é um super-conjunto do *Foundation Profile*, já o *Personal Profile* oferece algumas adições, como por exemplo, suporte ao uso de Xlets que é o pacote 1.1 (JSR 217) e a TV Digital esse perfil para sua configuração.

A figura 9 a seguir, mostra a estrutura geral da J2ME mostrando os Profiles a configuração.

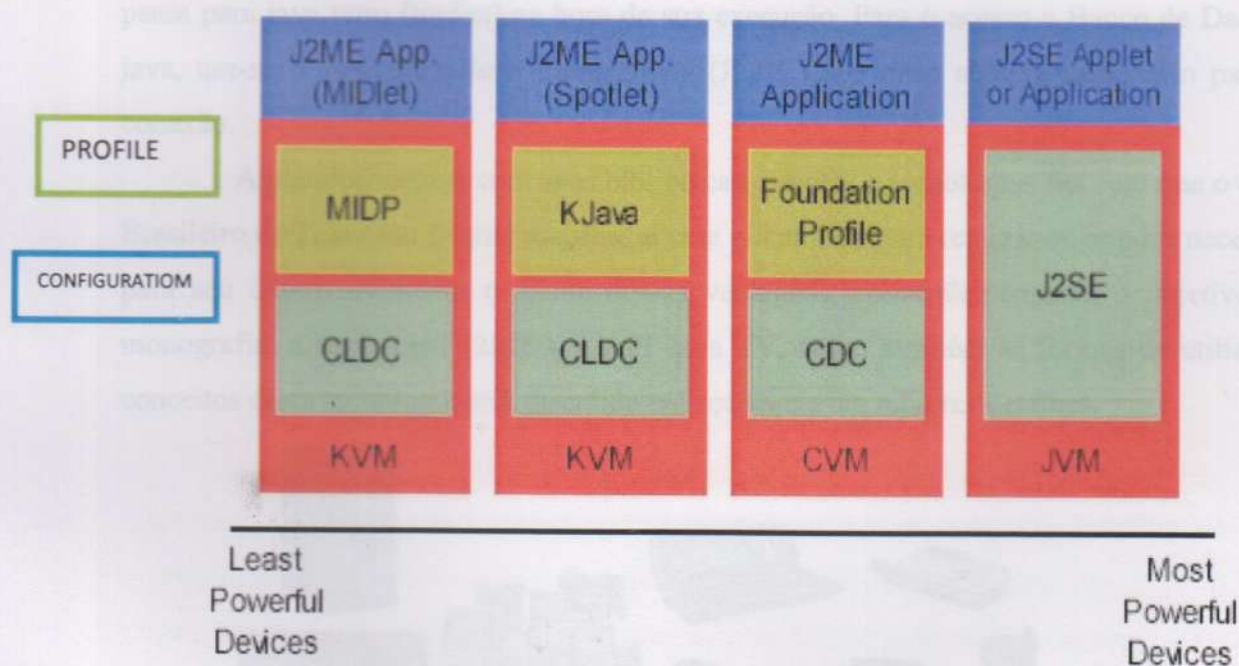


Figura 09: estrutura do perfil e da configuração dentro da J2ME (ABNT, 2008).

Outros pontos muito importantes na plataforma Java são:

- *Java Virtual Machine* (JVM) com várias bibliotecas destinadas à TV digital. Isso permite ao desenvolvedor escrever apenas uma única vez o código, pois a JVM torna compatível para os receptores, sem se preocupar em saber qual hardware e software usar,

é o que se chama de multiplataforma. Esta é a vantagem da plataforma java, pois torna a aplicação portátil e compatível.

- *Java Software Development Kit (JSDK)*, ou apenas *JDK*, é definido como o ambiente de desenvolvimento java que inclui o compilador, a *Java Virtual Machine*, os exemplos de bibliotecas Java e *appletviewer* para executar as *applets*.

- *Java Community Process (JCP)*, organização internacional de fabricantes e desenvolvedores com a responsabilidade de desenvolver e revisar as referências, especificações e tudo mais relacionado ao java. Para maiores informações, acesse www.jcp.org. Existe também a *Java Specification Request (JSR)*, que são documentos que propõem mudanças em algumas aplicações já existentes ou propostas para definições para implementações java.

- *Web Services Developers Pack (WSDP)*: API's e ferramentas para desenvolvimento de *WebServices* e *Servlets* em java. Para desenvolver aplicações Web, a plataforma java disponibiliza da tecnologia *Java Server Pages (JSP)*, linguagem de script que passa para java puro (traduz) na hora de sua execução. Para o acesso a Banco de Dados em java, usa-se o *Java DataBase Connectivity (JDBC)* que dá o suporte necessário para essa conexão.

A plataforma java com suas bibliotecas, pacotes e tecnologias fez com que o Comitê Brasileiro de Televisão Digital escolhesse essa plataforma para realizar os estudos necessários para seu desenvolvimento e diante dessas vantagens, conseqüentemente, o objetivo desta monografia, a tecnologia J2ME e a API Java TV, como também as formas de utilização e conceitos desta tecnologia será discutida em seqüência em maiores detalhes.



Figura 10 - edições da plataforma java 2 micro edition (KROLL, 2000).

As plataformas java contem a maioria dos recursos necessários para os sistemas receptores de TV digital facilitando o desenvolvimento de softwares uma vez que a linguagem java, que fornece uma plataforma de desenvolvimento para este tipo de equipamento, chamada de java TV, já é disponível. Essa plataforma contém um emulador que testa as aplicações mais facilmente e de maneira mais rápida.

Essa configuração fornece, segundo (Loureiro, 2004), estas funcionalidades:

- Fluxo de áudio e vídeo
- Acesso condicional
- Acesso aos dados nos canais de transmissão
- Acesso aos dados do *Service Information*
- Controle do sintonizador de canais
- Sincronização da mídia
- Gerenciamento do ciclo de vida das aplicações

2.7 JAVA TV

A java TV foi criada pela *Sun Microsystems* e desenvolvida no ambiente J2ME – *Plataforma Java 2 Micro Edition* (JAVA TV, 2005), sendo uma extensão da plataforma java. É uma API que utiliza o desenvolvimento de conteúdo para televisão digital interativa, porque provê as funcionalidades necessárias num receptor de TV digital.

A linguagem java foi uma escolha unânime entre o comitê brasileiro para se tornar a linguagem padrão do *middleware* ginga. O *middleware* é uma camada de software que facilita o desenvolvimento e a execução das aplicações interativas na televisão que ficou a cargo e responsabilidade da UFPB de desenvolver esse ambiente procedural enquanto a PUC - Rio ficou com a parte de que trata o sincronismo das mídias e adaptabilidade dos múltiplos dispositivos de exibição que tem como vantagem o tamanho pequeno de suas aplicações facilitando a transmissão da TV digital (SUN MICROSYSTEMS, 2003).

Utilizado, no set-top-box, a tecnologia java TV que é o ambiente de aplicação java com a API javaTV e um sistema operacional de tempo real, o programador consegue abstrair os detalhes de hardware e protocolos de comunicação existentes no set-top-box, já que o ambiente de aplicação deles é a *personal java*. Na figura 11, a uma comparação entre um set-top-box genérico com um utilizando a tecnologia javaTV.



Figura 11: arquitetura de um *set-top box* genérico e um utilizando a tecnologia Java (SUN MICROSYSTEMS, 2003).

Algumas funcionalidades da API java são os serviços de acesso e as informações de serviços que a API java trata tanto as programações interativas como as programações convencionais, todas como serviços. Há também o controle de canal de difusão que JavaTV utiliza o framework para gerenciar o conteúdo audiovisual de um *set-top-box*. Outra funcionalidade é o acesso aos dados de difusão que são conjuntos de seqüências de dados que são necessárias a uma aplicação que pode estar presente ou não na aplicação dando acesso às seqüências de dados a dados encapsulados por IP.

Por último, o gerenciamento do ciclo de vida de uma aplicação que tem seu início, suas mudanças de estados e por fim, sua destruição. Essa aplicação é chamada de Xlet e pode estar presente no *set-top-box* ou ser carregada e gerenciada pela API javaTV.

2.7.1 API Java TV

O desenvolvimento da API Java TV se deu em meados de 1998, lançando sua versão 1.0 em outubro de 2000. Seu principal objetivo é prover mecanismos de controle e acesso sobre receptores de TV Digital. Primeiramente, a API java era uma extensão da *Personal Java*, que hoje se encontra no conceito de *End Of Life* (EOL), ou fim da vida. A partir daí,

java TV passou a ser suportada pela família *Java 2 Micro Edition* (J2ME), com a configuração *Connected Device Configuration* (CDC), que oferece, no seu perfil mais amplo, quase toda as funcionalidades da J2ME nos dispositivos embarcados.

A API java tem funcionalidades específicas para os receptores de TV digital que não precisa de nenhuma outra API. As funcionalidades dessa API vão de acessos a serviços com gerenciamento total dos mesmos, a controle de gráficos e elementos de recepção de sinal de difusão, como também, a gerência nas aplicações. Um canal de TV é compreendido como um serviço utilizado no contexto de TV digital onde existem suas aplicações que viabiliza o conteúdo interativo. A tabela 01 mostra a arquitetura Java e seus componentes envolvidos.

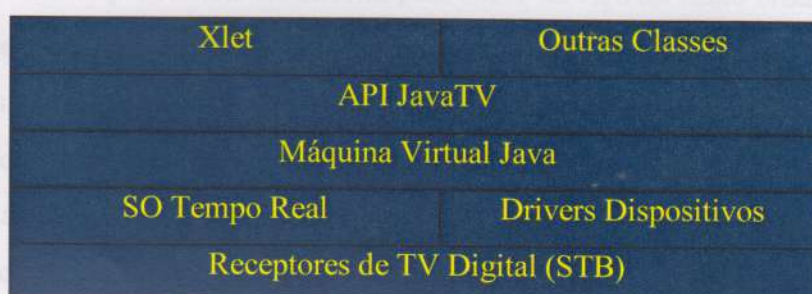


Tabela 01, Arquitetura Java TV (Revista mundo Java nº 17, 2008).

Uma aplicação para javaTV possui uma classe principal que implementa a interface Xlet e entre outras de uso específico que podem ser utilizados pela xlet, como também, padrões de *middleware* que usam outras classes. A xlet faz várias chamadas a API java que é executada sobre a máquina virtual java que por sua vez, é executada por um sistema operacional em tempo real que através de seus *drivers* acessa os dispositivos do receptor da TV digital. Java TV é integrado com ao *Java Media Framework* (JMF) que viabiliza a sincronização do conteúdo interativo com o áudio e vídeo de um programa em exibição (BATISTA, 2006).

2.7.2 Pacotes da Api JavaTV

Interface de Programação de Aplicativos (API) está organizada nos seguintes pacotes:

- `javax.tv.carousel`: acesso a arquivos de broadcast agregados em um sistema de arquivos no fluxo do vídeo;

- javax.tv.graphics: adiciona um suporte mínimo à biblioteca gráfica *Abstract Windowing Toolkit* (AWT), para solução de questões específicas em TV;
 - javax.tv.locator: Provê mecanismo para referências em formato de URL para serviços de broadcast e clipes de mídia broadcast;
 - javax.tv.media: extensões para suporte à integração JMF;
 - javax.tv.media.protocol: suporte JMF para protocolos de streaming broadcast;
 - javax.tv.net: provê mecanismos para acessar datagramas IP contidos em um fluxo broadcast;
 - javax.tv.service: descrição alto nível de serviços de TV digital, incluindo também mecanismos básicos para coleta de informações sobre serviço do fluxo broadcast;
 - javax.tv.service.guide: suporte para aplicações do tipo de guia eletrônico de programação (EGP) e seus conceitos associados (horário, classificação etária, etc.);
 - javax.tv.service.navigation: suporte à navegação em serviços de TV digital. Isso inclui suporte à lista de serviços favoritos, componentes agregados a serviços, etc.;
 - javax.tv.service.selection: conceitos descrevendo como os serviços são apresentados ao usuário e como um novo serviço pode ser selecionado. Há também a possibilidade de apresentação de múltiplos serviços de uma só vez (*Picture-in-picture*);
 - javax.tv.service.transport: conceito descrevendo os mecanismos de transporte para um serviço de TV digital;
 - javax.tv.util: classes utilitárias para aplicações javaTV, incluindo gerência de temporizadores e eventos temporizados;
- Um *Xlet* é uma aplicação java que mostra na TV digital a interatividade da programação. Um exemplo de aplicação é o *xi*, o *xi* é transmitido pelo canal de entrada pela TV.
- Pacotes da API HAVi utilizados pela especificação MHP e suas respectivas funções.
- org.dvb.application: fornece acesso a listas de aplicações que estão disponíveis neste contexto e a capacidade de lançar essas aplicações;
 - org.dvb.dsmcc: fornece acesso a arquivos transportados no fluxo transmitido;
 - org.dvb.event: fornece acesso às informações fornecidas pelo usuário acontecidos antes que eles sejam processados através do mecanismo de evento do pacote java.awt;
 - org.dvb.io.ixc: fornece suporte para a comunicação de interaplicações;

- org.dvb.io.persistent: fornece extensões para o pacote java para o acesso aos arquivos, realizada em armazenamento persistente;
- org.dvb.lang: fornece os elementos fundamentais relacionados com a plataforma que não foi encontrado no pacote Java.lang;
- org.dvb.media: fornece DVB a extensão específica para a Java Media Framework;
- org.dvb.net: fornece características gerais de rede que não foi encontrada em outro lugar;
- org.dvb.net.ca: fornece extensão para o acesso condicional API da DAVIC;
- org.dvb.net.rc: fornece sessão de gestão bidirecionais para IP e conexões que são baseadas em sessão do ponto de vista de uma aplicação;
- org.dvb.net.tuning: fornece extensões para a personalização da API de DAVIC;
- org.dvb.si: fornece acesso para informações de serviços DVB;
- org.dvb.test: Em um modelo de sistema Broadcast, existem efetivamente três principais entidades envolvidas em um processo automatizado: o teste servidor que é utilizada para realizar e dar início a todos os testes;
- org.dvb.ui: provê a funcionalidade de extensões gráficas;
- org.dvb.user: fornece acesso às configurações e preferências configuradas pelo usuário final.
- org.davic.media: oferece diversas extensões para o Java Media Framework para o controle do conteúdo de áudio e vídeo da TV (BATISTA, 2006).

2.8 XLET

Uma Xlet é uma aplicação java que mostra na TV digital a interatividade do computador convencional. Uma abstração da *applet* usada nas páginas html, o xlet é inserido no serviço da TV digital. A aplicação é identificada pelo *middleware* no ponto de entrada pela classe que implementa a interface `javax.tv.xlet.Xlet` e executa a aplicação usando a máquina virtual java. O *middleware* gerencia as aplicações com um componente que é responsável pelo ciclo de vida da aplicação chamando o método que muda o seu estado.

A figura 12 apresenta os estados de um xlet e os métodos responsáveis pelas transições.

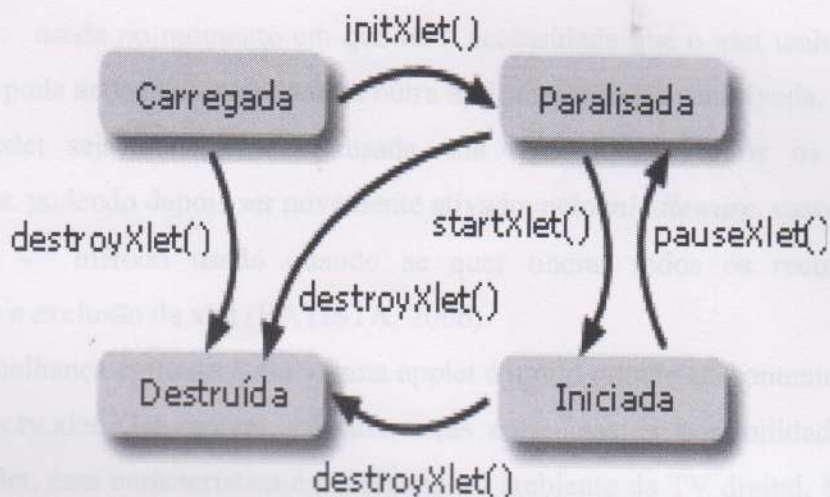


Figura 12, ciclo de vida de um Xlet (SUN MICROSYSTEMS, 2003).

As transições dos estados acontecem da seguinte forma:

- Carregado (*Loaded*) – neste estado, a xlet é carregada, a classe principal é instanciada pelo *middleware*, porém a xlet ainda não é inicializada, se houver alguma falha, a xlet passa para o estado destruído – *Destroyed*;
- Pausado (*Paused*) – a xlet é inicializada, mas não está utilizando nenhum recurso compartilhado do receptor, esse estado acontece quando o método *initXlet* é invocado no estado carregado ou quando a xlet é pausada (*pauseXlet*) no estado ativo;
- Ativo (*Active*) – estado onde a xlet está em funcionamento, isso acontece quando o método *startXlet* é invocado no estado pausado;
- Destruído (*Destroyed*) – acontece quando a xlet é destruída, liberando todos os recursos que tinha utilizado, o método *destroyXlet* pode ser invocado a partir de qualquer estado terminando seu funcionamento ou quando o estado de transição chega a exceção *XletStarteChangeException* que é lançada de qualquer transição de estados.

Os métodos invocados pela própria aplicação, no *middleware* ou pelo componente responsável, são chamados para que as mudanças de estados da aplicação aconteçam estão todos descritos na interface *Java.tv.xlet.Xlet*, são eles:

- *initXlet* – chamado pelo *middleware* para iniciar a xlet, passa o local de onde será executada a aplicação na forma de um objeto *xletcontext* que informa também ao *middleware* que a xlet mudou de estado, como mostra a figura 4, esse método igualmente como no estado de pausado, não utiliza nenhum recurso compartilhado e se houver alguma falha na chamada desse método, a exceção *XletStarteChangeException* é lançada e a xlet é destruída;
- *startXlet* – utilizada para informar que a xlet pode se tornar ativa ou se houver alguma falha, ser destruída, utiliza os recursos compartilhados para realizar suas funções;

- `pauseXlet` - usada no momento em que há a necessidade que o `xlet` tenha sua execução suspensa, isso pode acontecer quando uma outra aplicação precise ser ativada, por exemplo, e para que a `xlet` seja considerada pausada, ela deve liberar todos os seus recursos compartilhados, podendo depois ser novamente ativado, pelo *middleware*, caso necessite.
- `destroyXlet` - método usado quando se quer liberar todos os recursos utilizados relacionado-os a exclusão da `xlet` (BATISTA, 2006).

A semelhança entre uma `xlet` e uma `applet` é muito grande no contexto do método da interface `javax.tv.xlet.Xlet`, porém, há diferenças entre elas: a possibilidade de pausar a execução da `xlet`, essa característica é necessária no ambiente da TV digital. Um *set-top-box* pode executar várias `xlets` simultaneamente, mas, apenas uma `xlet` será visível, a qual usa os recursos necessários e as não visíveis ficam em estado pausado.

A `xlet`, em seu contexto normal, notifica ao *middleware* que houve mudança de estado, pois ela pode ser pausada por si própria ou se destruir. Quando a `xlet` é inicializada, o *middleware* passa como parâmetro, o objeto que implementa os métodos da `javax.xlet.XletContext` (BATISTA, 2006).

2.8.1 Xletview

É muito comum que um desenvolvedor tenha um ambiente simulado para que possa testar suas aplicações na TV digital, pois, não é todo mundo que tem uma rede de TV digital experimental à disposição, geralmente o ambiente é simulado com estações de teste ou com emuladores em software. Uma estação de testes inclui equipamentos e softwares que visam reproduzir todas as características relevantes em um ambiente de TV digital real. O emulador faz o papel do componente do *middleware* que gerencia as aplicações. Um emulador serve para que testes preliminares sejam feitos nas aplicações da TV digital, já que o ambiente possui vários elementos envolvidos, além do *middleware* (BATISTA, 2006).

No estado da Paraíba, na cidade de João Pessoa, a Universidade Federal UFPB, desenvolveu uma estação de software e hardware para testes, chamada de *Xtation*. O *xtation* é composto por equipamentos e programas que simulam o ambiente de TV Digital, incluindo transmissão de vídeo digital e seus elementos, sincronização de eventos, decodificação e execução de aplicações interativas, etc.

Um bom software para aprendizado e iniciação em desenvolvimento de aplicações para TV digital, é o emulador xletview (disponível em <http://xletview.sourceforge.net>). Ele foi desenvolvido para o teste de xlets compatíveis com o *middleware* MHP, possuindo parte da implementação das classes desse *middleware* que dão suporte à execução de xlets (incluindo parte da implementação dos componentes gráficos HAVI). Há também outros emuladores disponíveis como, por exemplo, a implementação criada pela Espial, possui as mesmas características do xletview podendo ser utilizada em qualquer plataforma.

Outra opção é o emulador mhp4free. Criado para ser executado pela plataforma Linux, é totalmente alemão, possui as implementações da api JavaTV e outras padrão MHP.'

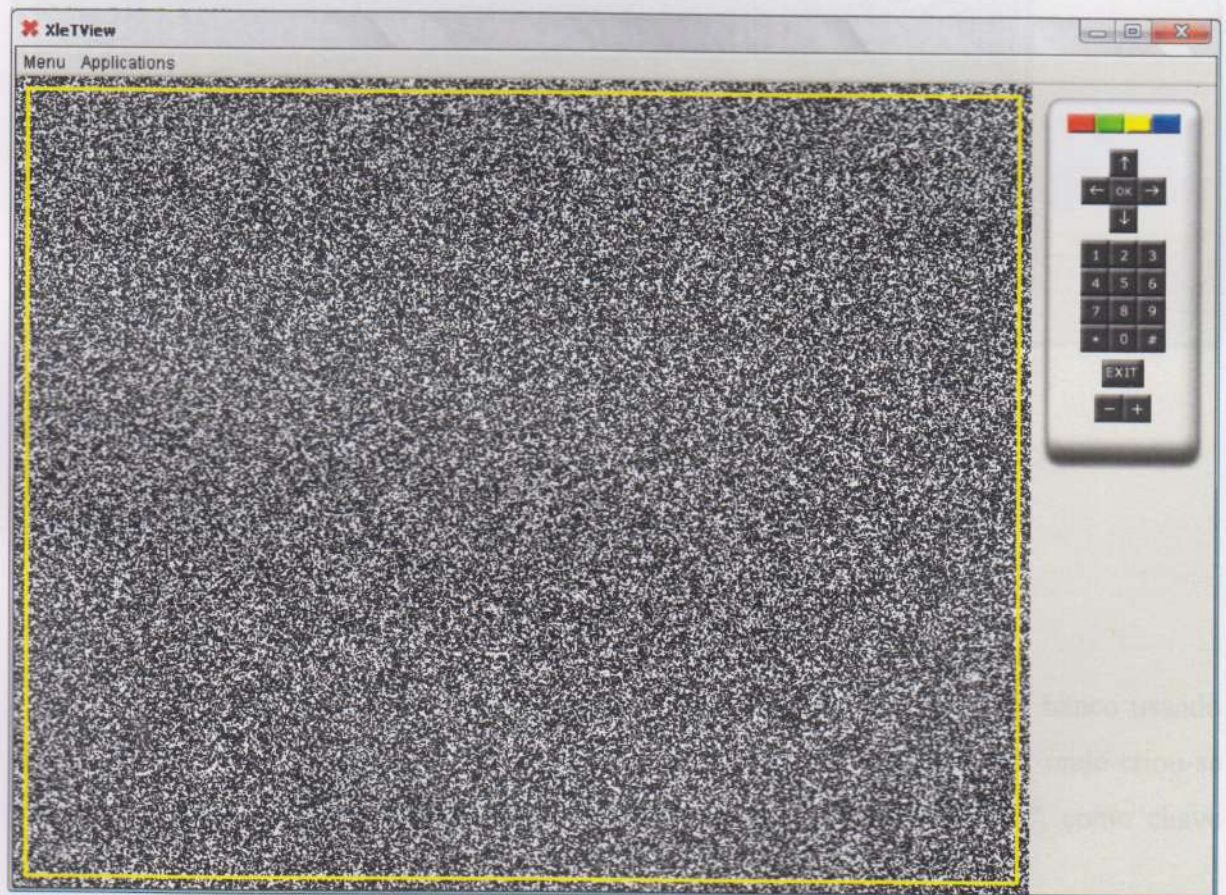


Figura 13: Tela inicial do Software Xletview (BATISTA, 2006).

O *xletview* possui botões padrão MHP que simulam um controle remoto onde os eventos podem ser capturados utilizando classes AWT ou HAVI, extensões de AWT (BATISTA, 2006).

Através do menu "*Application*", o submenu "*Manage Application*" aparecerá e lá é possível adicionar as xlets para execução e posteriormente agrupá-las também. Depois disso, a xlet entra no menu "*Application*" para que a execução dela seja iniciada.

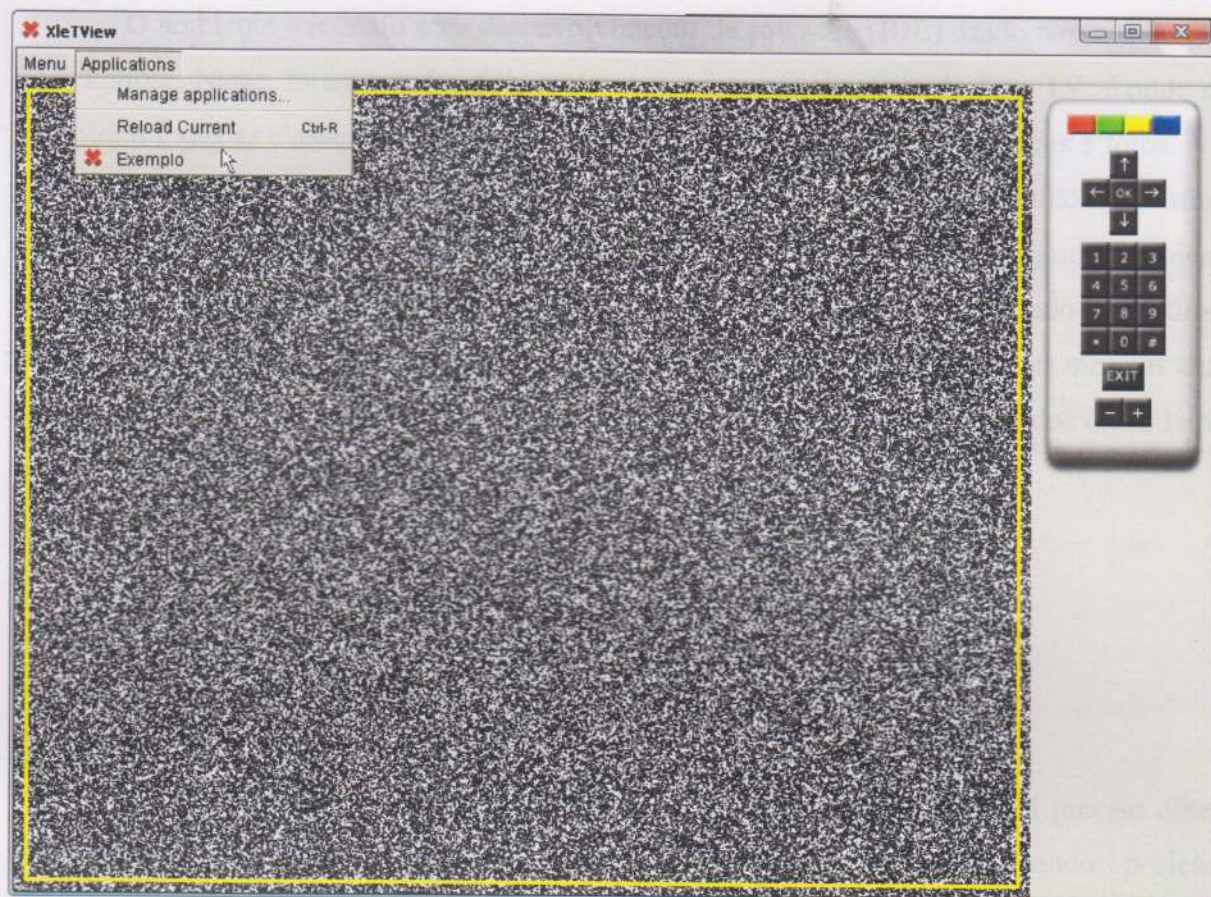


Figura 14: Aba de aplicações do emulador Xletview (BATISTA, 2006).

3 METODOLOGIA

Foi usada a tecnologia JDBC do java. A aplicação foi conectada ao banco usando esse mesmo *Servlet*. O banco de dados utilizado foi o *Postgres*, versão 1.6.3, onde criou-se uma tabela com três colunas: id, nome e quantidade de votos, tendo o “id”, como chave primária.

O código da servlet do projeto JavaTVServidor, que faz a conexão com o *Postgres* e os métodos de votar e recuperar voto para que seja mostrado os resultados no *ContainerResultado.java* do projeto principal javaTV (Anexo 1).

A aplicação foi desenvolvida em um Notebook particular, com a seguinte configuração: Notebook da marca Amazon, modelo L81 com processador Core Duo 1.6GHz, 2GHz de memória RAM, 128mhz de memória de vídeo compartilhada e 80GB de HD.

O ambiente integrado para desenvolvimento de software (IDE) usado nesse exemplo foi o Eclipse. Neste, foram criados dois projetos, o primeiro, chamado de Java TV, é onde se encontram todas as classes, métodos e arquivos usados na aplicação como vídeos e fotos. No pacote *Default* do projeto javaTV, existe 11 classes Java que fazem todo o trabalho de mostrar os botões, *Labels*, o menu, as fotos dos participantes e os vídeos. No segundo projeto encontra-se apenas uma *servlet* de conexão com o banco como já foi citado no último parágrafo. O projeto é organizado em vários contêiners, que são classes Java e as mesmas têm cada uma, sua função. Além dos contêiners, há também uma classe domínio dos candidatos onde são gravados seus atributos.

3.1 SINAL DIGITAL

Essa pesquisa científica tem caráter, de início, bibliográfica, pois, foi preciso obter conhecimento em livros e artigos sobre as tecnologias existentes tendo posição epistemológica positivista. A pesquisa tem, também, um caráter exploratório uma vez que foram pesquisadas as melhores ferramentas para desenvolver a própria aplicação, depois, tecnológica onde tratará conhecimentos já estudados na área, tecnologias existentes com o objetivo de obter novos processos para um novo produto. Terá ênfase no sinal da TV digital, entendendo como é feita a transmissão e recepção do mesmo acompanhando toda sua trajetória, mudanças de estados, formatos que serão usados e transformados ou mudados, os dispositivos de hardware envolvidos nos processos, os softwares que iram fazer o processamento do sinal para que ele chegue com maior qualidade de som e imagem nos receptores de cada telespectador que possua o serviço e, por fim, a principal característica que a TV digital possui: a interação do telespectador, a capacidade do usuário passar de passivo a ativo, enviando informações a TV através do controle remoto do aparelho de televisão por um canal de retorno que o padrão do sistema brasileiro de TV digital, no aspecto de transmissão e recepção, adotou para que esse processo tornasse realidade.

A pesquisa foi voltada, também, para o tipo de arquitetura que a TV digital possui, para que possa ser possível o entendimento do funcionamento do dispositivo receptor do sinal (*set-top-box*), é ele o responsável por todo o funcionamento da recepção do sinal decodificação e execução do *middleware* Ginga que irá fazer a parte de apresentação através do seu subsistema Ginga-NCL e o Ginga-J responsável pela interação. Dentro dele funciona a

plataforma java e sua API, responsável pela execução das xlets que irá fazer a interação com o usuário ser possível de ser realizada.

A proposta de estudo foi desenvolvida exclusivamente a partir de fontes já elaboradas. Conseguir tirar o máximo de conhecimento do orientador e visitar departamentos que já desenvolvam a tecnologia da TV digital no estado para obter maiores informações de como chegar ao objetivo dessa pesquisa que será o desenvolvimento de uma aplicação na TV digital que usará a interatividade como destaque.

Para o desenvolvimento da aplicação, será usado as API's da javaTV que serão de suma importância para o caminhar da aplicação já que sem elas, isso não seria possível. Essa aplicação será feita em cima do pacote J2ME que é específico para o desenvolvimento de aplicações de dispositivos móveis onde se utiliza a configuração do tipo CDC que suporta as funcionalidades da tecnologia Java mesmo estando em aparelhos móveis. Essa configuração dá suporte também à execução de xlets dos Set-top-boxes. No CDC, existem três principais perfis, o que será usado é o *Personal Profile*, que é o mais completo entre os outros dois existentes.

Uma vez executada a xlet, ela vai fazer várias chamadas a API java que será executada sobre a máquina virtual java e por sua vez é executado por um sistema operacional em tempo real onde seus drives vão fazer a conexão com os receptores da TV digital. O *Java Media Framework* (JMF) faz a sincronização entre a parte interativa de áudio e vídeo de um programa em exibição e a tecnologia JDBC fará a conexão com o banco de dados.

3.2 APLICAÇÃO

“Tanto nas engenharias como na computação são necessários estudos anteriores para que se possa determinar como está funcionando o processo ou como é o processo, para que posteriormente seja recomendado, desenvolvido e implantado um novo sistema, produto ou técnica de otimização” (JUNG, 2004). Na aplicação, foi feita algumas xlets sendo executadas com vídeos onde ao mesmo tempo em que o mesmo é apresentando, quadros serão mostrados ao seu lado com a possibilidade do usuário, através do controle remoto, poder selecionar algum quadro que seja apresentado e a partir daí, será mostrado alguma informação relacionada ao vídeo como, por exemplo: compras, estatísticas, mudança de vídeo, enquetes, chats, etc.

O emulador que será usado é o *Xletview 3.0.6* da SUN Microsystem, ele será responsável por simular no desktop ou notebook, o ambiente da TV Digital. O *xletview* se conectará a IDE Eclipse e executará a *xlet* como se fosse ao aparelho de televisão digital. Para nós desenvolvedores, é muito mais rápido para realizar os testes necessários nas aplicações usando o emulador, porém, antes de tudo é preciso instalar a IDE de preferência (Eclipse ou Netbeans), as bibliotecas *javaTV* necessárias para sua aplicação e por fim, o emulador.

Com essa aplicação voltada para a TV digital, o telespectador terá, através do seu controle remoto, todas as características de um computador, no seu aparelho de televisão, por isso, a aplicação terá também a parte de acesso a dados que serão gravadas em banco de dados usando o *JDBC* do *java*, quando necessário. Ícones serão mostrados na tela de emulação junto com o vídeo e cada ícone mostrado terá sua função ou link no seu código que levará a outra tela onde será possível o usuário navegar e executar ações como no computador normal.

Figura 12: Emulador Xletview com a aplicação em execução (Dados da pesquisa, 2004).

3.3 RESULTADOS E DISCUSSÃO

A aplicação foi desenvolvida com sucesso e o resultado foi bastante interessante. A maior dificuldade foi decomprimir o arquivo de vídeo que foi executado na aplicação, pois, o

A aplicação é uma simulação de um programa de televisão onde o apresentador faz uma enquete. Durante a transmissão do programa, o telespectador, através do controle remoto, poderá participar de uma eleição que perguntará ao usuário quem será o mais votado na enquete “Bola Cheia” e “Bola Murcha”, no caso, o telespectador votará em qual vídeo ele mais gostou para o “Bola Cheia” e o “Bola Murcha”, terá também a possibilidade de navegar pelo menu proposto, visualizando o resultado da enquete e ainda a tela de informações e estatísticas sobre os participantes concorrentes. A figura 15, mostra a interface do *xletview* rodando a aplicação.

As ferramentas para simulação citadas no decorrer do texto, a que se mostrou melhor, é o *xletview*. Esse emulador leva extrema vantagem em relação de facilidade de configuração e a implementação das *API's* abrindo um leque maior, em relação aos outros, de possibilidades no desenvolvimento de *xlets*.

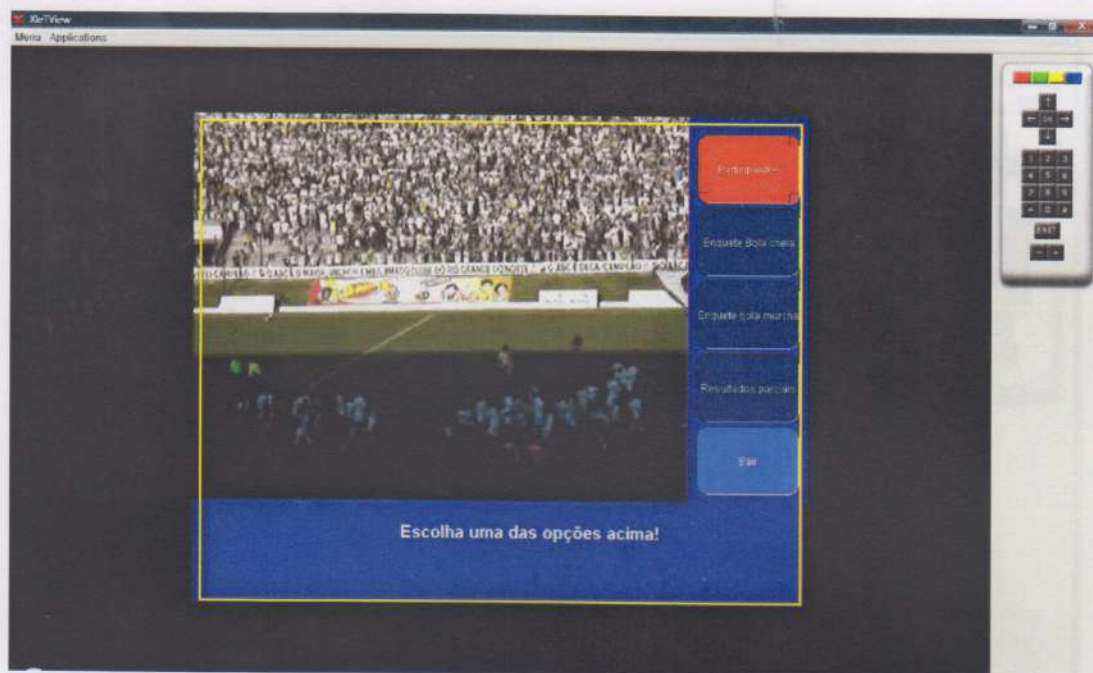


Figura 15: Emulador Xletview com a aplicação em execução (Dados da pesquisa, 2008).

A aplicação foi desenvolvida com sucesso e o resultado foi bastante interessante. A maior dificuldade foi descomprimir o arquivo de vídeo que foi executado na aplicação, pois, o java media Framework não aceita arquivos de vídeo comprimidos, por isso, foi preciso usar os Codecs adequados de cada vídeo para que os mesmos pudessem ser exibidos. A partir daí, a conexão com o banco de dados foi feita a partir de um novo projeto que foi chamado de JavaTVServidor, onde foi criada uma *servlet*.

A figura 16 mostra o xletview com a aplicação em andamento e a primeira tela esperando uma ação do usuário, quando é pressionado o botão vermelho, a xlet é executada e o menu principal é mostrado.

Comparando as ferramentas para emulações citadas no decorrer do texto, a que se sobressai melhor, é o xletview. Esse emulador leva extrema vantagem em relação da facilidade de configuração e a implementação das Api's abrindo um leque maior, em relação aos outros, de possibilidades ao desenvolvedor de xlets.

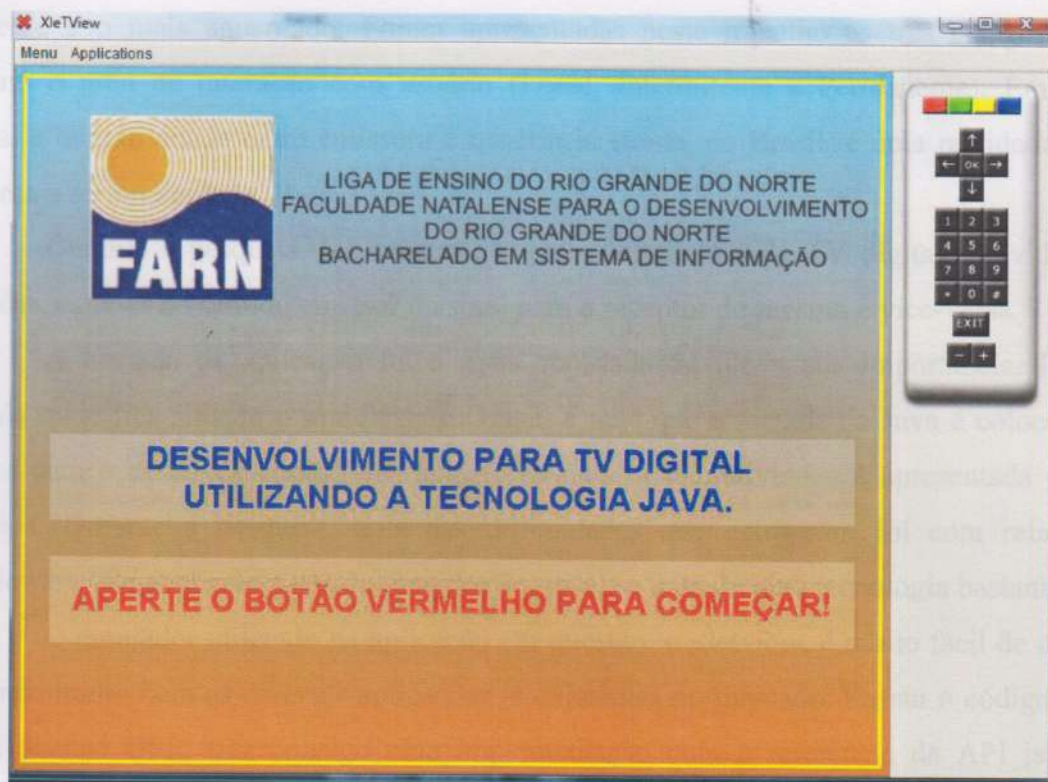


Figura 16: Emulador Xletview com a aplicação em esperando uma ação do usuário (Dados da pesquisa, 2008).

4 CONCLUSÃO

Este trabalho analisou conhecimentos já estudados na área de Sistemas de Informação e as tecnologias existentes no mercado, com ênfase no sinal da TV digital. Com base nos resultados obtidos, concluiu-se que a cada dia a televisão digital vem sendo progressivamente implantada nas principais capitais brasileiras. A implantação mais recente ocorreu na cidade de Salvador no estado da Bahia, cidade que inclusive foi a primeira capital do nordeste a ser implantado o sistema digital de televisão. Nos próximos semestres, novas cidades empregarão o sistema digital de televisão e assim sucessivamente. Baseando-se nesse cenário, é possível afirmar com clareza o interesse por parte do governo federal e em breve, das empresas de tecnologia de um modo geral.

Conclui-se que a tecnologia da TV digital está criando um novo ambiente para a tecnologia, onde novas oportunidades e desafios estão surgindo. É importante compreender os conceitos técnicos, econômicos, políticos e sociais que envolvem esse ambiente. A interatividade, se não for o maior, pode ser um dos fatores mais importantes desse novo

ambiente e o mais aguardado. Foram apresentadas neste trabalho as três categorias que formam o grau de interação com usuário (Local, Intermitente e Permanente). Em outros países, a interatividade entre emissora e usuário já existe, no Brasil, é uma realidade e está próxima a ser implantada.

Durante a pesquisa, foi constatado que a arquitetura da TV digital é dividida em camadas, estudas no sentido emissor do sinal para o receptor do mesmo e vice-versa.

A camada de aplicação foi a mais abordada devido a sua importância. É nessa camada onde fica situado o middleware Ginga, é nele que a tecnologia Java é colocado em prática para o desenvolvimento das aplicações onde a interatividade é apresentada para os usuários. Durante a pesquisa, uma das dificuldades que ocorreram foi com relação ao middleware Ginga quanto a sua documentação, pois, se trata de uma tecnologia bastante nova.

O emulador utilizado na aplicação em questão, o xletview, é muito fácil de operar e responde muito bem as diversas aplicações já existentes no mercado. Possui o código aberto sob a licença GPL, traz consigo uma implementação com a referência da API javaTv e também de outras API's especificadas no padrão MHP, como a Havi (Home Audio-Video Interoperability), DAVIC (Digital Audio-Video Council) e implementações especificadas pela DVB, além das bibliotecas do personal java que o mesmo faz uso.

O xletview é programado totalmente em Java. Pode ser executado tanto em uma plataforma Linux quanto no Windows, bastando para isso utilizar o Java 2 standard development kit para compilar as xlets e executar o xletview. Porém, o mesmo é adequado para aplicações simples, quando é usado em aplicações mais complexas, o Xletview deixa a desejar por não ter muita qualidade em rodar vídeos com maior qualidade. Outra deficiência desse emulador é a incapacidade de exibir vídeos relacionados ou controlados por uma xlet. O que pode ser feito é rodar o vídeo diretamente pelo emulador e colocar imagens para a apresentação desse mesmo vídeo.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **TV Digital** (<http://www.abnt.org.br/default.asp?resolucao=1280X1024>). Acesso em dezembro 2008.

BATISTA, CARLOS. **TV Digital**. Curitiba - PR. REVISTA MUNDO JAVA, Edição 17, Ano III, Editora Mundo, 2006.

BECKER, V., MONTEZ, C. **TV digital interativa: conceitos, desafios e perspectivas para o Brasil**. 1. ed. Florianópolis: I2TV, 2004.

DAVENPORT, D. **Net4TV Voice: First Look at AOLTV**. 2000. Disponível em: <http://www.net4tv.com/voice/story.cfm?storyid=3056>. Acesso em: 12 nov. 2008.

DVB SERVICES. **Digital video broadcasting services**. Disponível em: <http://www.dvbservices.org>. Acesso em: 5 abr. 2005.

FERNANDES, J.; LEMOS, G.; SILVEIRA, G. **Introdução à televisão digital interativa: arquitetura, protocolos, padrões e práticas**. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 24., 2004, Salvador. Disponível em: <http://www.cic.unb.br/docentes/jhcf/MyBooks/itvdi/texto/itvdi.pdf>. Acesso em: 12 nov. 2008.

FREEVO. **User Gallery**. Disponível em: <http://freevo.sourceforge.net>. Acesso em: 12 nov. 2008.

GARLAN, D. **Software Architecture: a Roadmap**, In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 22., 2000, Limerick. Disponível em: <http://www.cs.ucl.ac.uk/staff/A.Finkelstein/fose/finalgarlan.pdf>. Acesso em: 12 nov. 2008.

GINGA DF. **Arquitetura do middleware**. Disponível em: <http://gingadf.blogspot.com/2009/04/o-que-e-java-dtv-tem-custo-ou-nao.html> />. Acesso em: 10 dez 2008.

JUCÁ, P., LUCENA, U. **Experiências no desenvolvimento de aplicações para televisão digital interativa**. In: FORUM DE OPORTUNIDADES EM TELEVISÃO DIGITAL INTERATIVA, 3., 2005, Poços de Caldas. Disponível em: <http://www.inf.pucpcaldas.br/eventos/tvdi2005/VersaoFinalPaperTVDI2005PaulyneJuca.pdf>. Acesso em: 12 nov. 2008.

JUNG, CARLOS FERNANDO. **Metodologia para pesquisa e desenvolvimento**. Rio de Janeiro: Axcel Books, 2004.

KROLL, MICHAEL; HAUSTEIN, STEFAN. **Java 2 Micro Edition (J2ME) Application Development**. Editora Pearson, 2000.

LOUREIRO, JANINE DE AGUIAR. **Interface de Programação para o Desenvolvimento de Aplicações para TV Digital**. Universidade de Pernambuco. Trabalho de Conclusão 2004. Disponível em www.cin.ufpe.br.

MACIEL, R. P., ASSIS, S. R. **Middleware: uma solução para o desenvolvimento de aplicações distribuídas**. Revista CienteFico, Salvador, ano IV, v 1, jan. 2004.

MIDDLEWARE GINGA-NCL. **Documentos, tutoriais e artigos**. Disponível em: WWW.ncl.org.br. Acessado em: 09 de set.2008.

MIRANDA, CLÁUDIO. **Dados em J2ME**. Curitiba- PR. Java Magazine, edição 03,

Editora Neoficio, 2002.

OPEN GINGA. **Desenvolvimento e pesquisa, publicações**. Disponível em :
WWW.openginga.org. Acessado em : 12 de Nov. 2008.

SOFTWARE PÚBLICO BRASILEIRO. **Middleware Ginga**. Disponível em:
<http://www.softwarepublico.gov.br/spb/ver-comunidade?community_id=1101545>
Acessado em 27/05/2008.

SUFRAMA. **TV Digital Interativa**. Disponível em:
<http://www.suframa.gov.br/download/documentos/programa_tv_digital.pdf>. Acesso em: 12
nov. 2008.

SUN MICROSYSTEMS. **O que é Java?**. Disponível em: <
http://www.java.com/pt_BR/download/whatis_java.jsp>. Acesso em: 26/09/2008.

SUN MICROSYSTEMS. **Tutoriais**. Disponível em
<<http://java.sun.com/products/cldc/wp/KVMwp.pdf>> acessado em 26/05/2008.

TIOBE, **Programming Community Index**. Disponível em: <
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acessado em: May
2008.

ANEXO 1

ANEXO I - Código da Servlet(javaTVServidor) do projeto(colocar de onde tirei), página 51.

```

1 package servidor;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class for Servlet: EnqueteServlet
7  */
8 public class EnqueteServlet extends javax.servlet.http.HttpServlet implements
9     javax.servlet.Servlet {
10     static final long serialVersionUID = 1L;
11
12     /*
13      * (non-Java-doc)
14      *
15      * @see javax.servlet.http.HttpServlet#HttpServlet()
16      */
17     public EnqueteServlet() {
18         super();
19     }
20
21     /*
22      * (non-Java-doc)
23      *
24      * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request,
25      *     HttpServletResponse response)
26      */
27     public static Connection getConnection() {
28         Connection con = null;
29
30         String url = "jdbc:postgresql://localhost:5432/TvDigital";
31         try {
32
33             Class.forName("org.postgresql.Driver");
34
35             con = DriverManager.getConnection(url, "postgres", "admin");
36         }
37     }
38 }

```

```

46
47     } catch (ClassNotFoundException e2) {
48         System.err.println("Não achou o driver");
49
50     } catch (SQLException e1) {
51         System.err.println("Falha ao conectar");
52     } finally {
53
54         System.out.println("Conexão aberta. ");
55     }
56
57     return con;
58 }
59
60 protected void doGet(HttpServletRequest request,
61     HttpServletResponse response) throws ServletException, IOException {
62
63     int operacao = 0;
64     String operacaoStr = request.getParameter("operacao");
65     if (operacaoStr != null && operacaoStr.length() > 0)
66         operacao = Integer.parseInt(operacaoStr);
67
68     int idVoto = 0;
69     String votoStr = request.getParameter("idVoto");
70     if (votoStr != null && votoStr.length() > 0)
71         idVoto = Integer.parseInt(votoStr);
72
73     switch (operacao) {
74         // votar
75         case 1:
76             votar(idVoto);
77             break;
78
79         // recuperar quantidade de votos
80         case 2:
81             response.getOutputStream().print(recuperarVoto(idVoto));

```

```

2         break;
3     }
4 }
5
6 }
7
8 /*
9  * (non-Java-doc)
10  *
11  * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request,
12  *      HttpServletResponse response)
13  */
14 protected void doPost(HttpServletRequest request,
15     HttpServletResponse response) throws ServletException, IOException {
16     // TODO Auto-generated method stub
17 }
18
19 private void votar(int idVoto) {
20
21     Connection con = geConnection();
22     try {
23         String sql = "UPDATE encuesta set qtdVoto = (qtdVoto+1) where id = ?";
24
25         PreparedStatement ps = con.prepareStatement(sql);
26         ps.setInt(1, idVoto);
27
28         ps.executeUpdate();
29     } catch (Exception e) {
30         e.printStackTrace();
31     }
32 }
33
34 private int recuperarVoto(int idVoto) {
35     int qtdVoto = 0;
36     Connection con = geConnection();
37     try {
38         String sql = "SELECT qtdVoto from encuesta where id = ?";
39
40         PreparedStatement ps = con.prepareStatement(sql);
41         ps.setInt(1, idVoto);
42
43         ResultSet rs = ps.executeQuery();
44         if (rs != null && rs.next()) {
45             qtdVoto = rs.getInt("qtdVoto");
46         }
47     } catch (Exception e) {
48         e.printStackTrace();
49     }
50     return qtdVoto;
51 }
52 }

```