

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE ESPECIALIZAÇÃO EM DESENVOLVIMENTO DE SISTEMAS
COM TECNOLOGIA JAVA**

**APLICATIVO DE TV DIGITAL: ABORDAGEM DAS TECNOLOGIAS E
DESENVOLVIMENTO DE UM PROTÓTIPO**

TRABALHO DE CONCLUSÃO

EVANDRO DE FREITAS GONÇALVES JÚNIOR

JUIZ DE FORA, MINAS GERAIS, BRASIL

2011

**APLICATIVO DE TV DIGITAL: ABORDAGEM DAS TECNOLOGIAS E
DESENVOLVIMENTO DE UM PROTÓTIPO**

por

Evandro de Freitas Gonçalves Júnior

Trabalho de conclusão apresentado à Universidade Federal de Juiz de Fora
para obtenção do grau de
Especialista em Desenvolvimento de Sistemas com Tecnologia Java

JUIZ DE FORA, MINAS GERAIS, BRASIL

2011

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE ESPECIALIZAÇÃO EM DESENVOLVIMENTO DE SISTEMAS
COM TECNOLOGIA JAVA**

A Comissão Examinadora Abaixo Assinada,
Aprova o Trabalho de Conclusão:

**APLICATIVO DE TV DIGITAL: ABORDAGEM DAS TECNOLOGIAS E
DESENVOLVIMENTO DE UM PROTÓTIPO**

Elaborado por

EVANDRO DE FREITAS GONÇALVES JÚNIOR

Como requisito parcial para a
obtenção do grau de
Especialista em Desenvolvimento de Sistemas com Tecnologia Java

Prof. Eduardo Barrére (Orientador)

Prof. Marcelo Ferreira Moreno

Prof. Tarcísio de Souza Lima

JUIZ DE FORA, MINAS GERAIS, BRASIL

2011

AGRADECIMENTOS

Agradeço a Deus pela força e por me dar condições de realizar esse trabalho.

A minha família e amigos que estiveram do meu lado me auxiliando nessa caminhada, tornando as dificuldades mais brandas.

A minha mãe pela lição de superação que me foi muito valiosa e me permitiu ajustar os meus parâmetros de dificuldade.

Ao meu pai por acreditar no meu potencial e tornar possível mais essa etapa na minha vida acadêmica.

A minha namorada, Ana Cristina, por me incentivar e nunca me deixar desistir dos meus sonhos.

*“Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;*

*Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear
Though as for that the passing there
Had worn them really about the same,*

*And both that morning equally lay,
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads onto way,
I doubted if I should ever come back.*

*I shall be telling this with a sigh
Somewhere ages and ages hence;
Two roads diverged in a wood, and I –
I took the one less traveled by,
And that has made all the difference.”*

Robert Frost, 1916

RESUMO

GONÇALVES JR, Evandro de Freitas. **Aplicativo de tv digital: abordagem das tecnologias e desenvolvimento de um protótipo**. Trabalho de Conclusão de Curso (Pós Graduação em Desenvolvimento de Sistemas com Tecnologia Java). Universidade Federal de Juiz de Fora, Juiz de Fora, 2011.

A TV Digital está se difundindo cada vez mais no âmbito mundial. Foi seguindo esse crescimento que o Brasil criou seu próprio padrão de TV Digital, o Ginga, após analisar os padrões vigentes em outros países. O presente trabalho aborda as principais tecnologias envolvidas no ambiente Ginga e mostra como criar uma aplicação imperativa que seja executada na TV Digital brasileira, utilizando o módulo Ginga-J e a linguagem Java – Adotada como padrão para o desenvolvimento de aplicações imperativas – e que consiga reproduzir uma videoaula.

Palavras-chave: TV Digital. Ginga. Ginga-J. Java.

ABSTRACT

Digital Tv is increasingly spreading worldwide. It was following this growth that Brazil created its own digital TV standard, Ginga, after reviewing the standards used in other countries. This paper addresses the key technologies involved in Ginga environment and shows how to create an imperative application that runs on the Brazilian Digital Tv, using the Ginga-J module with Java language - Adopted as standard for the development of imperative applications - and that can play video lessons.

Keywords: Digital Tv. Ginga. Ginga-J. Java.

LISTA DE FIGURAS

FIGURA 1	Java, suas plataformas e seus grupos de hardware.....	7
FIGURA 2	Camadas da plataforma JME.....	9
FIGURA 3	Arquitetura do middleware Ginga.....	20
FIGURA 4	APIs que compõem o Ginga-J.....	22
FIGURA 5	Configurando protótipo no emulador Ginga-J.....	29
FIGURA 6	Protótipo rodando no emulador Ginga-J.....	30

SUMÁRIO

INTRODUÇÃO	1
1 METODOLOGIA	4
1.1 ENQUADRE TEÓRICO	4
2 JAVA	6
2.1 JAVA MICRO EDITION	8
2.2 JAVA TV	10
3 TELEVISÃO	12
3.1 TV DIGITAL	14
3.2 TELEVISÃO E INTERATIVIDADE.....	16
3.3 TV INTERATIVA NO BRASIL SE FAZ COM GINGA.....	18
3.4 GINGA-J.....	21
4 PROTÓTIPO	24
4.1 TECNOLOGIAS UTILIZADAS	24
4.2 CONFIGURAÇÃO DO EMULADOR E DESENVOLVIMENTO.....	25
CONCLUSÃO	31
REFERÊNCIAS	34
ANEXO I – CLASSE XLETVIDEO	40
ANEXO II – CLASSE MANAGER	42

INTRODUÇÃO

A televisão é um eletrodoméstico altamente difundido entre a população brasileira. Segundo dados da Pesquisa Nacional por Amostra de Domicílios (Pnad) realizada em 2009, 95,7% dos domicílios tinham aparelho de televisão (IBGE, 2009).

Existem vários conceitos para Televisão, um deles é o que podemos observar segundo a enciclopédia eletrônica Wikipedia “Televisão (do grego tele - distante e do latim visio - visão) é um sistema eletrônico de reprodução de imagens e som de forma instantânea. Funciona a partir da análise e conversão da luz e do som em ondas eletromagnéticas e de sua reconversão em um aparelho”.

A televisão, a partir desse ponto será mencionada como Tv, está passando por uma significativa mudança, processo semelhante ao ocorrido quando as transmissões começaram a serem feitas regularmente em cores no ano de 1954, nos Estados Unidos da América. Hoje, a grande mudança que presenciamos é a troca do sistema analógico para o digital.

O início da transmissão digital abre novas possibilidades para a criação de conteúdo diferenciado e interativo. Com ela, a Tv deixa de ser um meio de comunicação de mão única, normalmente feito de uma emissora para muitos telespectadores, para permitir a interação dos usuários, transmissão de conteúdos em alta definição (*High Definition* ou *HD*), transmissão de conteúdos múltiplos e simultâneos no mesmo canal, entre outras facilidades. O que possibilita esse processo de interação é a maior capacidade de tráfego de informações no meio digital e a existência de um canal de retorno na Tv, como um modem interno ligado a uma linha telefônica ou a uma rede *wireless*.

A Tv Digital possui um canal de dados e até quatro canais de vídeo/áudio o que possibilita a transmissão simultânea de conteúdos diferentes para o aparelho de Tv. Isso se dá, por exemplo, com um subcanal que pode ser utilizado para transmitir o vídeo e áudio de um jogo de futebol, enquanto outro subcanal pode transmitir informações sobre a escalação dos times, dos jogadores, dos técnicos, e ainda outro canal é capaz de transmitir um quiz sobre o time com perguntas e respostas que podem ser respondidas através do controle remoto de sua Tv.

Há quem acredite que toda essa interatividade além de tornar a Tv mais atraente para quem assiste, também pode auxiliar no processo de inclusão digital de famílias de classe social com poucos recursos, uma vez que permite interação com o conteúdo exibido, muito semelhante ao que ocorre ao utilizarmos o computador (PINTO, QUEIROZ-NETO, SILVA, 2007).

É pensando nessa interatividade e no grande número de pessoas interessadas em Tv que esta pesquisa visa abordar a utilização de sistemas web na Tv Digital através do Ginga-J (IBGE, 2009).

Ginga é o nome dado para o *middleware* adotado no padrão da Tv Digital Brasileira, Sistema Nipo-Brasileiro de Tv Digital Terrestre (ISDB-TB)¹, desenvolvido em conjunto pelos laboratórios Telemídia da PUC-Rio e LAViD da UFPB.

O Ginga-J é um subsistema do Ginga que utiliza Java, que, segundo Yoshida & Franco, é “uma linguagem de programação inovadora que foi introduzida no mercado em 1995 pela Sun Microsystems” (2004, p. 10). “Ginga-J provê uma infraestrutura de execução de aplicações Java e extensões especificamente voltadas ao ambiente de TV.” (PUC-Rio/UFPB).

Sendo assim, o estudo pretende mostrar os passos necessários para criar um aplicativo *Xlet* que seja executado e consiga reproduzir uma videoaula utilizando o *middleware* Ginga em um emulador que simule o ambiente de Tv Digital. Outra pretensão é apontar as dificuldades encontradas durante o processo e mostrar as soluções utilizadas para resolvê-las.

¹ Disponível no sítio eletrônico: <http://www.ginga.org.br/pt-br/sobre>

1 METODOLOGIA

O modelo proposto visa construir um *Xlet* capaz de reproduzir uma videoaula, executado em um emulador simulando o ambiente de Tv Digital nos padrões do Sistema Brasileiro de Tv Digital Terrestre.

São abordadas as principais tecnologias envolvidas no processo de criação de um aplicativo nesse ambiente que servirá como aplicação dos conceitos e como base para identificar os problemas encontrados permitindo concluir as falhas no emulador, as dificuldades de construção e configuração dos aplicativos. Ele consiste no desenvolvimento de uma aplicação em Java que é executada em um emulador simulando o ambiente de Tv Digital com o *middleware* Ginga, e que reproduz um vídeo em formato *Audio Video Interleave (AVI)*, escolhido por ser compatível com o modelo adotado pelo Ginga, contendo uma videoaula de matemática.

1.1 ENQUADRE TEÓRICO

O ímpeto de realizar este estudo surge das novas possibilidades abertas pelo início das transmissões utilizando a tecnologia digital no Brasil e sua expansão para todo o território nacional com o tempo. Como essa tecnologia ainda não está totalmente difundida o número de pesquisas relacionadas a ela ainda é pequeno.

Pretende-se com o presente estudo contribuir com a comunidade de desenvolvimento de aplicativos para Tv Digital Brasileira, abordando um tema interessante e de grande utilidade tendo em vista que um dos pontos altos do padrão brasileiro de Tv Digital é a difusão do conhecimento e a gratuidade das tecnologias envolvidas, pensadas para estimular o maior número possível de pessoas a gerar conteúdo interativo. Espera-se que o protótipo desenvolvido possa servir como base para o desenvolvimento de aplicações no ambiente de Tv Digital que consigam reproduzir vídeos no formato *AVI*, formato suportado pelo padrão adotado no Brasil, uma vez que essa utilização de aplicativos educativos no suporte ao ensino é um novo caminho para a forma de produzir conteúdos para a Tv.

2 JAVA

Na empresa *Sun Microsystems* nasceu um projeto chamado *Green Project*, liderado por James Gosling, Patrick Naughton e Mike Sheridan. Com um pensamento visionário, seu objetivo era criar ferramentas para promover a convergência de aparelhos eletroeletrônicos. Esse grupo de profissionais considerava que “*O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a “próxima onda” do mundo digital.*” (MORAES, 2009).

Para provar a viabilidade de sua ideia, em 1992, eles criaram um protótipo chamado de *7 (*Star Seven*) que consistia em um controle remoto com tela sensível ao toque e um ajudante virtual, batizado de Duke, que ensinava o usuário a utilizar o aparelho. A função do dispositivo era a de controlar diversos aparelhos e aplicações, porém, para que fosse possível criar o *7, James Gosling desenvolveu uma nova linguagem de programação voltada para o aparelho e a chamou de *Oak*. Ao desenvolver o projeto, os líderes do grupo observaram algo importante conforme pode ser percebido na citação

abaixo:

Uma das primeiras conclusões durante o processo de desenvolvimento foi a de que o sistema não poderia ficar preso a plataformas, pois os clientes não estavam interessados no tipo de

processador que estavam utilizando, e fazer uma versão do projeto para cada tipo de sistema seria inviável (DIAS, FONTES, PIMENTEL, 2009).

No ano de 1995, Gosling lançou uma nova versão da linguagem *Oak* adaptando-a para a Internet e a chamou de Java. Com o passar dos anos a tecnologia Java vem crescendo de maneira acelerada. Considerando que ela nasceu para funcionar entre dispositivos heterogêneos e, por isso, vem sendo amplamente utilizada na Internet, a Java já atingiu a impressionante marca de 4 milhões de desenvolvedores em todo o mundo no ano de 2003, segundo Moraes mestre em computação e pesquisador da tecnologia Java.

Atualmente, a tecnologia Java se subdivide em plataformas e cada uma delas é voltada para um grupo de hardware. São elas: Java Standard Edition (JSE), Java Enterprise Edition (JEE) e Java Micro Edition (JME) conforme detalha a figura abaixo:



FIGURA 1 - Java, suas plataformas e seus grupos de hardware.

Fonte: <http://www.devmedia.com.br/post-6484-CONCEITOS-BASICOS-DAS-PLATAFORMAS-JAVA-E-J2ME.html> Artigo escrito por: Cristiano Fioresi Andreão

Para o presente estudo utilizaremos a plataforma Java Micro Edition para desenvolver o protótipo, uma vez que ela é designada para dispositivos com pouco poder de processamento, e o Set-top-box da TV Digital se enquadra nesse grupo.

2.1 JAVA MICRO EDITION

A Java Micro Edition, a partir de agora chamado de JME, é uma plataforma da tecnologia Java criada para atender dispositivos móveis e com pouca capacidade de processamento. Porém para que isso fosse possível, foi necessário reduzir alguns recursos da linguagem e também da Java Virtual Machine, conforme esclarecido na citação abaixo:

Visando fornecer os recursos necessários ao desenvolvedor, mas tendo em vista as limitações computacionais, foram criadas Application Programming Interfaces (API's) responsáveis por prover recursos específicos e que são utilizadas pelo programador apenas quando necessárias. (COSTA, 2009)

Assim como a tecnologia Java, a JME também é independente de hardware e é dividida em camadas conforme pode ser observado na figura 2.

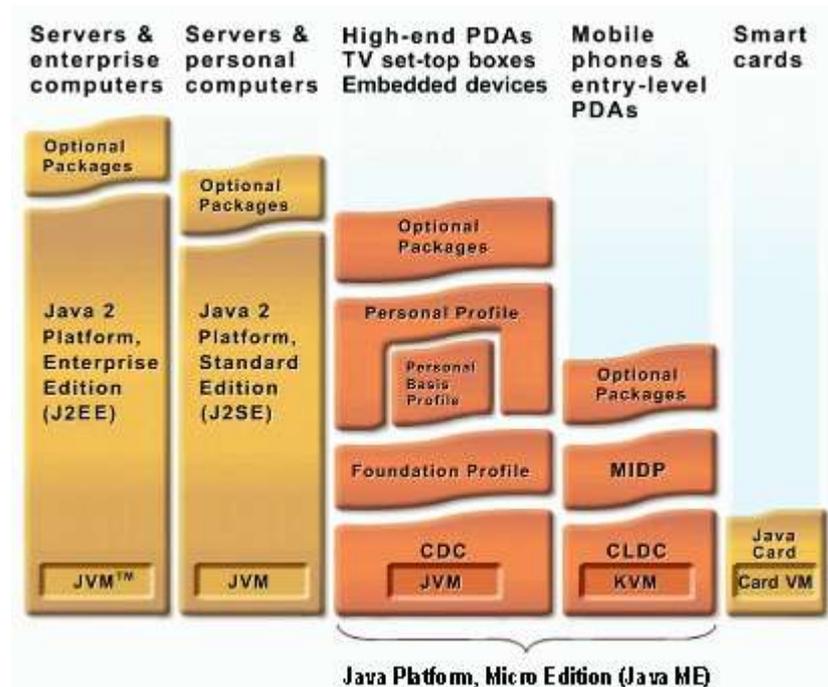


FIGURA 2 - Camadas da plataforma JME.

Fonte: <http://www.oracle.com/technetwork/java/javame/java-me-overview-402920.html>

Analisando as camadas podemos observar na base a máquina virtual. Ela varia de acordo com a configuração a ser utilizada. No caso da *Connected Device Configuration* (CDC) a máquina virtual é a *Java Virtual Machine* (JVM), já para a *Connected Limited Device Configuration* (CLDC), é utilizada a *Kilobyte Virtual Machine* (KVM). Essa distinção de configurações se dá devido à capacidade de recursos computacionais como memória e processamento. O CDC abrange dispositivos com mais recursos como smartphones, navegadores GPS de automóveis e set-top-box de TV Digital. Já no grupo do CLDC estão dispositivos mais limitados como os celulares e PDAs.

Os perfis estão logo acima da camada de configuração. Eles podem ser o *Foundation Profile* para o CDC e o *Mobile Information Device Profile* (MIDP) no caso do CLDC. Por último ficam os pacotes opcionais que são APIs criadas para auxiliar no desenvolvimento de aplicações.

Uma das APIs desenvolvidas para auxiliar na criação de programas para TV Digital é a Java TV que será melhor detalhada no próximo subcapítulo.

2.2 JAVA TV

Criada pela empresa multinacional dos Estados Unidos que fornece soluções em *hardware* e *software*, a *Oracle*, a API Java TV é uma tecnologia baseada na JME que oferece aos desenvolvedores um conjunto de soluções seguras, fáceis de implementar e de boa performance para construção de aplicações Java que rodem em ambiente de TV Digital ou em set-top-boxes. Ela também provê aos desenvolvedores acesso às principais características comuns das diferentes implementações de receptores.

Os telespectadores que tiverem receptores digitais habilitados para Java poderão receber e interagir com as aplicações Java TV enquanto assistem aos programas de TV e a ferramenta utilizada nessa interação é o controle remoto da televisão.

O nome dado às aplicações Java TV é Xlet. Como os applets, os Xlets são controlados pelo software que os roda. No caso de um Xlet, o software subjacente é o receptor de televisão digital ou set-top box que suporta a plataforma Java TV. Outra característica dos Xlets é que eles não têm o método *main()* e eles sempre implementam a interface *Xlet*. Essa interface define a assinatura dos métodos responsáveis pelo controle do ciclo de vida do *Xlet*. Estes métodos são responsáveis por sinalizar as seguintes mudanças de estado do *Xlet*:

- Create
- Initialize
- Start
- Pause
- Destroy

Toda implementação da Java TV tem um gerenciador de aplicação responsável por chamar os métodos de controle de ciclo de vida para mover um ou mais *Xlets* através dos estados listados acima. Porém, a interface não provê a codificação dos métodos, sendo o desenvolvedor responsável por criar tais implementações de acordo com a aplicação em questão determinando o que acontece com o *Xlet* em cada ponto de seu ciclo de vida.

O próprio *Xlet* pode iniciar algumas mudanças de estado por conta própria e depois informar ao gerenciador de aplicação sobre tais mudanças através da invocação do método *XletContext* da interface. Ao fazer isso, um objeto do tipo *XletContext* é passado para o *Xlet* em questão, quando ele é inicializado, permitindo a ele uma forma de recuperar propriedades e informar mudanças internas de estado.

A API Java TV consiste em classes e interfaces que permitem o processamento e manipulação de vídeos, áudio e dados enviados para o receptor digital através do sistema de difusão realizado pelas emissoras de tv. O receptor digital recebe o vídeo, áudio e os dados e os processa pelo pipeline de dados transmitindo o resultado para o aparelho de TV.

Alguns pacotes da Java TV são `Javax.tv.carousel`, `Javax.tv.graphics`, `Javax.tv.locator`, `Javax.tv.media`, `Javax.tv.media.protocol`, `Javax.tv.xlet`.

3 TELEVISÃO

Em 1817, após a descoberta do selênio pelo sueco Jacob Berzelius, deu-se início ao que seria a invenção da televisão, pois em 1873, o inglês Willoughby Smith conseguiu provar que tal elemento químico tinha a propriedade de transformar a energia elétrica em energia luminosa e assim se pensou que era possível transmitir imagens por meio de corrente elétrica. Mas as transmissões só viriam a ocorrer de fato em 1920, com experimentos de britânicos e americanos (PÁSCOA, 2006).

Sendo assim, poderíamos definir televisão como um sistema eletrônico de transmissão instantânea de imagens e sons que são comumente projetados em aparelhos receptores. Mas, sabemos que há muito mais a ser dito sobre essa tecnologia, uma vez que sua importância não se limita à transmissão de informações.

Muitas transmissões marcaram época e aproximavam o mundo do que viria a ser a chamada globalização. Um dos marcos da televisão foi em 1937, quando 50 mil espectadores acompanharam por três câmeras a coroação de Jorge VI, na Inglaterra e logo em seguida a televisão chegaria à Rússia e aos

Estados Unidos. É válido ressaltar que a nova tecnologia teve um desaceleramento nas produções devido a II Guerra Mundial, mas no período pós-guerra o mundo foi literalmente invadido pelo novo meio de comunicação (cf. Enciclopédia Britânica do Brasil, v. 14, 1998).

Os avanços da televisão não cessariam, passaram a ocorrer transmissões de cunho político, discursos de presidentes, coroação da rainha da Inglaterra, e transmissões de entretenimento, como as esportivas a partir de 1954, e até a primeira viagem espacial em 1961, seria transmitida. Logo, a televisão seria aprimorada passaria a ser em cores e surgiria a *tv a cabo* e *tv por satélite*, a DBS, sigla de *Direct Broadcasting System* o que trouxe melhor qualidade as transmissões, além de serviço exclusivo àqueles que pagariam uma mensalidade para obter a nova tecnologia (cf. Enciclopédia Britânica do Brasil, v. 14, 1998).

Sobre a história da televisão, sabe-se que foi chamada por muitos de “quinta parede dos aposentos” (Enciclopédia Britânica 1998, p. 31). O termo não é ultrapassado, pois a cada dia surgem mais avanços deste aparelho midiático e a televisão ganha mais espaço nas residências. Um exemplo disso é o sistema HDTV, *High Definition Television*, televisão em alta definição. O pioneiro nessa tecnologia foi o Japão em 1989. Sabe-se que os receptores de HDTV apresentam o dobro de resolução vertical e horizontal dos demais televisores. E estima-se que a qualidade da imagem seja semelhante à dos filmes usados no cinema (MELO, RIOS, GUTIERREZ, 2000).

A televisão é até hoje um dos meios de comunicação com grande inserção no mundo, e por sua vez é um instrumento importante para a cultura de massa, pelo modo objetivo que atinge um público e pela facilidade de determinar comportamentos sociais.

Foi pensando nessa disseminação e na possibilidade de interação do telespectador com o conteúdo da tv digital, interação essa que será melhor explicada no próximo subcapítulo, que surgiu um novo modo de ensino o *T-Learning* (JOKIPELTO, 2005), que consiste no uso da tv digital como meio de instrução através de requisições de vídeo aulas, *quiz*, exercícios e outros tipos de conteúdo feitos para serem atraentes para o aluno e educativos ao mesmo tempo.

E é com isso em mente que o presente estudo visa criar um protótipo de um sistema que rode no ambiente de tv digital e que possibilite a requisição pela Internet, através do controle remoto da televisão, de vídeo-aulas e seus roteiros.

3.1 TV DIGITAL

A Tv está passando por uma mudança significativa de paradigma, processo semelhante ao ocorrido quando as transmissões começaram a serem feitas regularmente em cores no ano de 1954, nos Estados Unidos da América, que é a troca do sistema analógico para o digital.

O funcionamento da transmissão analógica ocorre da seguinte maneira: Uma câmera capta a imagem de uma cena a uma taxa de 30 quadros por segundo, no próximo passo ela transforma a imagem em fileiras de pontos individuais chamados pixel, cada pixel recebe uma intensidade e uma cor. Após isso, ocorre uma combinação das fileiras utilizando sinais de sincronismo horizontal e sincronismo vertical e essas informações são transmitidas de

maneira que a eletrônica do televisor saberá como mostrar as fileiras de pixels em sua tela.

Segundo o professor e mestre em tv digital, Maurício Donato,

Na transmissão analógica que trabalhamos com o sistema PALM, nós temos 525 linhas de transmissão, mas [...], a nossa televisão varrendo de cima para baixa da esquerda para direita, só tem a capacidade de enxergar 480 linhas ativas. Isso é uma primeira questão da tv analógica. Já na tv digital nós subimos essa qualidade quase que seis vezes mais, tanto na transmissão quanto na recepção, aonde na transmissão a captação dentro das televisões, a gente trabalha com câmeras já desenvolvendo trabalhos em linhas de 1920 x 1080 e transmitimos em 1080 linhas. E na nossa recepção atualmente em tv digital, nos temos a condição de já receber também as 1080 linhas (DONATO, 2008).

A maior diferença desses números pode ser contemplada principalmente na qualidade e fluidez das imagens em movimento, uma vez que a quantidade de pixels utilizados na formação das imagens é muito superior no sistema digital.

Outro fator que sofreu uma melhora significativa foi o sistema de áudio. Enquanto no modelo analógico era possível utilizar mono ou estéreo, um canal de som ou dois canais, esquerda e direita, com a tv digital é possível a transmissão e recepção de 5.1 surround, que proporciona uma qualidade de som de cinema (DONATO, 2008).

A transmissão digital possibilita a criação de novos conteúdos, interativos e inovadores, modificando a forma como a Tv é utilizada hoje. Ela deixa de ser um veículo de mão única que apenas exhibe o conteúdo da emissora aos telespectadores, passando a permitir interação dos usuários com seu conteúdo

através do controle remoto. Suas inovações não se atêm apenas a interação, elas vão muito além, com a exibição de conteúdos em alta definição (*High Definition ou HD*), transmissão de conteúdos múltiplos e simultâneos no mesmo canal, entre outras (HORA *apud* CRUZ, 2008, p. 17).

Toda essa inovação só é possível graças a maior capacidade de tráfego de informações no meio digital e a existência de um canal de retorno na televisão, como um modem interno ligado a uma linha telefônica ou a uma rede wireless.

3.2 TELEVISÃO E INTERATIVIDADE

Se há alguns anos atrás nos fosse dito que seria possível pela televisão acessar uma conta bancária, interagir com amigos em redes sociais e ainda jogar videogame com alguém em outro ponto do planeta, provavelmente estaríamos pensando em um filme de ficção científica. Mas com a interação entre as mídias isso tudo se tornou possível e é o que vem acontecendo em total realidade. Segundo especialistas em produção como a analista de desenvolvimento de produtos da Panasonic Bruna Hatakeyama, *“Atualmente a gente [Panasonic] tem cerca de 70% do nosso portfólio conectado a internet. Para 2011, até o final do ano, a gente vai ter 100% das TVs, todo mundo quer estar conectado.”* (SEVERIANO, 2011).

Em outras palavras, há quem acredite que a programação da televisão ficou em segundo plano, pois os aparelhos oferecem muito mais aos seus

consumidores que agora se colocam no comando do que desejam usufruir da televisão.

A programação convencional da televisão não vem mais satisfazendo um público exigente, daí o acesso à rede visa solucionar esse impasse. Assistir televisão passa a ser um conceito ativo, o espectador pode assistir somente o que lhe interessa. E esse casamento entre Tv e internet promete ser fundamental para as futuras produções de televisores. E assim passará a existir no mercado aparelhos que apresentam qualidade de som e imagem e que permitem a interatividade.

Sabe-se também que essa tecnologia ainda não apresenta um preço acessível à população, e que por isso pode prejudicar a popularização desse recurso no Brasil. Outro fator que deve ser levado em conta é o acesso à internet e a velocidade do sinal que pode variar de acordo com a intencionalidade do uso e que ainda é precário em muitas localidades nacionais.

Mas mesmo diante desses obstáculos o mercado se encontra otimista, pois a interatividade associada à televisão traria novamente o conceito de a televisão ser um ponto de encontro da família.

É pensando nisso que está surgindo uma nova abordagem de desenvolvimento de aplicações voltadas para a Tv Digital, como a criação de novos tipos de propaganda, envolvendo métodos diferenciados dos praticados atualmente, programas com acesso a serviços do governo, sistemas que permitem marcação de consultas, tele-aulas interativas, etc (CAPDA, 2004).

Esses conteúdos podem ser tratados de uma forma inovadora no ambiente de Tv Digital, melhor do que as já praticadas nos computadores atualmente.

3.3 TV INTERATIVA NO BRASIL SE FAZ COM GINGA

Na escolha pelo padrão de transmissão de tv digital a ser adotado pelo Brasil foram estudados os três modelos já estabelecidos no mundo: O americano, *Advanced Television System Committee* (ATSC-T), o japonês, *Integrated Services Digital Broadcasting Terrestrial* (ISDB-T), e o europeu, *Digital Video Broadcasting* (DVB-T).

Após analisarem as vantagens e desvantagens de cada um deles, os técnicos responsáveis optaram por adotar o sistema japonês porque ele era o único que oferecia portabilidade gratuita, alta definição e possibilitava a criação de conteúdos interativos, além de permitir ao Brasil criar seu próprio *middleware*², o Ginga³, para sustentar seu próprio padrão. Sendo assim:

O ISDB-TB (*International System for Digital Broadcasting-Terrestrial*), adaptado do sistema Japonês (ISDB-T), e também denominado de sistema Nipo-Brasileiro, se configurou como o padrão de transmissão de TV Digital no Brasil, tendo como vantagem perante os outros modelos além do uso de um *middleware* exclusivo, a adoção de um padrão de compressão de vídeo muito mais moderno e eficiente: o MPEG-4 ou H.264, conforme observa Mizukosi (2008), além do AAC

² *Middleware* é o nome dado para uma camada de tecnologia que fica entre o hardware e os aplicativos do sistema. Ele é responsável por prover mecanismos para execução das linguagens oferecidas e bibliotecas para auxiliar no desenvolvimento de aplicações.

³ A escolha foi Ginga, que é o nome dado ao movimento base da capoeira, como forma de homenagear o povo brasileiro no seu modo peculiar de ser, assim como sua cultura e busca por igualdade.

para compressão de áudio. (ZANCANARO, SANTOS, TODESCO, 2009).

O ISDB-T é muito versátil porque além de permitir enviar sinais de TV digital também pode ser utilizado em várias atividades como na transmissão de dados, como receptor parcial em celulares ou dispositivos PDAs ou também em computadores, no acesso a sites da Internet, na atualização do receptor e também para sistemas multimídia visando fins educacionais.

Por essas e outras qualidades ele foi escolhido para ser utilizado como padrão brasileiro, porém com algumas modificações como a substituição do padrão MPEG2 pelo MPEG4, mais novo e completo, além do acréscimo de tecnologias desenvolvidas em nosso território.

A seguir podemos verificar mais detalhes sobre os ajustes realizados pelo Brasil no padrão ISDB-T:

Em Junho de 2006, o governo brasileiro anunciou que o ISDB-T seria a base para modulação. No entanto, a codificação de vídeo seria H.264, também chamado "MPEG-4 parte 10" e áudio "HE-AAC v2" também conhecido como AAC+, enquanto no Japão é utilizado o MPEG-2 para o serviço de vídeo em alta definição e áudio MPEG-2 Layer 3. Isso explica o "B" inserido após o "T" de ISDB-TB, que foi incorporado para diferenciar o padrão brasileiro do japonês. (SIERVI, 2010)

O Ginga surgiu de um trabalho conjunto dos laboratórios Telemídia da PUC-Rio e LAViD da UFPB, e foi totalmente construído utilizando tecnologias livres de royalties e abertas para estimular a participação popular na produção de conteúdos interativos para nosso ambiente de tv.

Ele está dividido em três módulos o Ginga Common Core, o Ginga-NCL e o Ginga-J conforme a figura 3.

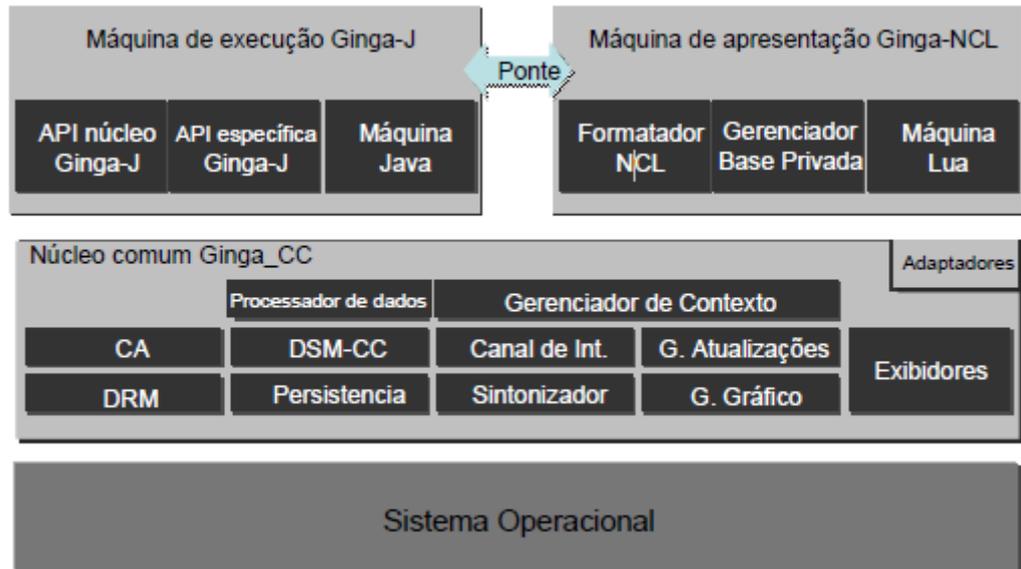


FIGURA 3 – Arquitetura do *middleware* Ginga.

Fonte: http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_06_soares.pdf
 Artigo escrito por: Luiz Fernando Gomes Soares

O módulo Ginga Common Core, Ginga-CC, é responsável por prover as bibliotecas comuns a todas as aplicações desenvolvidas, tanto declarativas para o Ginga-NCL, quanto imperativas para o Ginga-J. Essa característica possibilita boa portabilidade do modelo uma vez que basta adaptar o módulo Ginga-CC para o local que ele será embarcado para que as aplicações já existentes continuem funcionando sem necessidade de modificações.

O módulo Ginga-NCL é responsável por realizar o processamento e prover artefatos de desenvolvimento para aplicações declarativas *Nested Context Language* (NCL) (SOARES, 2009). O NCL e a linguagem Lua formam a base para a criação de programas declarativos para o Sistema Brasileiro de Tv Digital Terrestre.

Por sua vez, o módulo Ginga-J é responsável por realizar o processamento e prover artefatos de desenvolvimento para aplicações imperativas utilizando a linguagem Java.

Os subsistemas lógicos Ginga-J e Ginga CC apresentados acima serão utilizados no desenvolvimento do protótipo proposto no presente estudo.

3.4 GINGA-J

Ginga-J é o núcleo procedural do middleware Ginga criado tendo como base as tecnologias Java e agrega muitas inovações através de APIs, porém mantém a característica de portabilidade inerente ao Java, sendo compatível com a maioria dos middlewares atuais de TV digital. Ele deverá obedecer as especificações da norma NBr 15606-4 descrita pela Associação Brasileira de Normas Técnicas (ABNT).

O padrão digital brasileiro nasceu tendo a conversão digital como obrigatória já que ele foi pensado para promover a transmissão do sinal digital para diversos tipos de dispositivo como as televisões normais e de alta definição além de aparelhos celulares e outros dispositivos móveis. O Ginga-J prevê também a capacidade de comunicação entre dispositivos utilizando as tecnologias Wi-Fi, infravermelho, Bluetooth, Ethernet ou qualquer outro tipo de tecnologias comumente utilizadas nas redes atualmente.

O Ginga-J é composto por três conjuntos de *APIs* chamadas Verde, Azul e Amarelo, conforme figura abaixo.

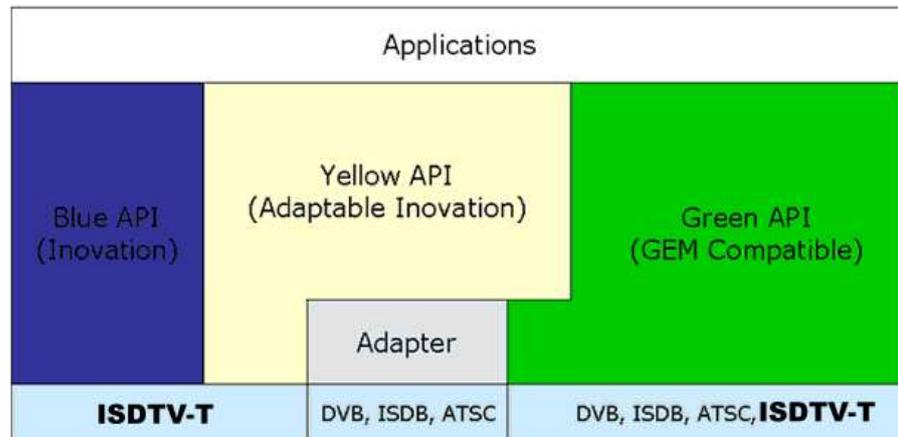


FIGURA 4 – APIs que compõem o Ginga-J.

Fonte: <http://bibliotecadigital.sbc.org.br/download.php?paper=625>

Artigo escrito por: FILHO, LEITE, BATISTA

As APIs da Ginga Verde são compatíveis com a GEM⁴, e inclui as APIs Java TV, DAViC e HAVi, disponíveis também em outros padrões de TV digital. As da Ginga Amarela são extensões propostas para completar as necessidades específicas do Brasil e que podem ser implementadas pelo uso de programas adaptadores da Ginga Verde. Dentro da Ginga Amarela estão, entre outros, o *Java Media Framework (JMF)*, que é um *framework* necessário para o desenvolvimento de aplicações mais complexas, como, por exemplo, alguma que tenha captura de vídeo ou áudio. E por fim, as Azuis são aquelas não compatíveis com a GEM, porém oferecem meios de integração de diferentes dispositivos de entrada e saída para o Set-top box e permite a integração de conteúdos declarativos e procedurais na mesma aplicação. Sendo assim, as aplicações desenvolvidas utilizando as APIs Ginga Verde podem ser executadas no *middleware* Ginga normalmente além de poderem também ser executadas em qualquer *middleware* que atenda as especificações GEM. As feitas com as APIs Ginga Amarela somente conseguem ser executadas pelo *middleware* Ginga se o adaptador necessário for transmitido e executado junto com a aplicação. Já as aplicações feitas com as APIs Ginga Azul somente

⁴ “*Globally Executable Middleware*, ou GEM, é uma especificação de *middleware* aberto da *Digital Video Broadcasting* que permite a implementação de aplicações interativas através de redes de difusão e de banda larga, bem como para discos *Blu-Ray*”. Conforme documento no sítio eletrônico: http://www.dvb.org/technology/fact_sheets/DVB-GEM_Factsheet.pdf

serão executadas em ambientes com *middleware* Ginga (FILHO, LEITE, BATISTA, 2006).

4 PROTÓTIPO

O protótipo desenvolvido de uma aplicação de tv digital utilizando Ginga-J e tecnologia Java nos moldes do padrão brasileiro SBTVD, visa aplicar os conceitos levantados nos capítulos anteriores.

Este capítulo tem a finalidade de apresentar as tecnologias utilizadas na construção do protótipo, assim como imagens das telas dele em funcionamento.

4.1 TECNOLOGIAS UTILIZADAS

Para desenvolver o protótipo foi necessário seguir passos rigorosos na configuração do ambiente de programação para Ginga-J e do emulador de TV Digital, ambos disponibilizados pelo Laboratório de Aplicações de Vídeo Digital (LAViD) da Universidade Federal da Paraíba⁵.

⁵ Disponível no sítio eletrônico <http://gingacdn.lavid.ufpb.br/projects/gingaj-emulador/wiki/Baixando_o_c%C3%B3digo_do_reposit%C3%B3rio>.

No processo de desenvolvimento foram utilizados, além do emulador supracitado, a Integrated Development Environment (IDE) Eclipse⁶ Indigo e a Java Software Development Kit versão 1.6.

4.2 CONFIGURAÇÃO DO EMULADOR E DESENVOLVIMENTO

Os passos realizados para a execução do emulador foram:

1. Baixar o código do Emulador Ginga-J através do Git⁷ Hub utilizando o comando:

```
git clone git://dev.openginga.net/gingaj-emulador.git
```

Feito isso, o código do emulador será salvo no computador na pasta pré-definida durante a instalação do Git Hub;

2. Para executar o emulador é necessário abrir seu código-fonte no Eclipse. Para isso, abra a IDE, e clique no menu “*File >> Import >> Existing Projects into Workspace*”. Na caixa de busca que será aberta, procure a pasta onde foi salvo o código do emulador Ginga-J salvo no passo 1 e clique em “*Finish*”. Agora o projeto está disponível no *Package Explorer*;
3. Na versão atual do emulador, suas dependências vêm pré-ajustadas assim que o projeto é importado para a IDE, portanto não é necessário configurá-las manualmente;

⁶ Eclipse é uma *IDE* que permite a programação, entre outras linguagens, em Java. Foi desenvolvida inicialmente pela empresa IBM e depois cedida como software livre para a comunidade. Disponível para download em < <http://www.eclipse.org/>>.

⁷ Git, segundo a enciclopédia digital Wikipédia é “um sistema de controle de versão distribuído com ênfase em velocidade”, comumente utilizado no desenvolvimento de aplicações de código aberto.

4. Para executá-lo é necessário clicar com o botão direito do mouse sobre o projeto e escolher a opção “*Run As >> Java Application*”. Será exibida uma caixa com todos os métodos executáveis existentes no projeto, escolha a classe “*Main*” do pacote “*net.beiker.xletview*”. Ao realizar esse passo, ocorrerá um erro. Para corrigi-lo, clique com o botão direito do mouse sobre o projeto e escolha as opções “*Run As >> Run Configurations*”.
5. Na tela exibida, escolha a opção “*Java Application >> Main*” no menu esquerdo, depois a aba “*Classpath*”. Em seguida, clique em “*User Entries*” e no botão “*Advanced*”. Selecione “*Add Folders*” e escolha a pasta com o nome do projeto do emulador salvo no computador.

Após esses passos descritos acima, o emulador estará pronto para ser executado. Ele é peça fundamental no desenvolvimento de aplicações para TV Digital uma vez que tem o papel de simular um ambiente de TV nos padrões Ginga-J, permitindo assim, os testes e execuções das aplicações criadas.

Vale ressaltar que o emulador não está totalmente concluído e ainda contém muitas falhas de desenvolvimento e funcionamento, o que pode acarretar no mau funcionamento, ou no não funcionamento total, das aplicações nele testadas. Prova disso foi a iniciativa da empresa Intacto TV Digital que lançou um projeto chamado OpenITVLab⁸ que se propõe a ser um canal de auxílio aos desenvolvedores de aplicativos para a TV Digital brasileira.

Nele o desenvolvedor envia seu aplicativo para a equipe da empresa pelo site do OpenITVLab, depois disso a equipe testa a aplicação via ar em vários conversores digitais de diferentes marcas, da mesma maneira que uma

⁸ Disponível em: <<http://www.intacto.com.br/OpenITVLab/>>.

emissora de TV faria. Eles gravam os resultados dos testes, enviam para o autor e disponibilizam no canal do *Twitter*⁹ @TVDigitalBrasil.

Um dos testes existentes no canal é de uma aplicação que veio como exemplo no próprio ambiente de desenvolvimento padrão disponibilizado pelo LAViD chamada Ginga Hero. O programa é executado de maneira correta no emulador porém quando submetido aos testes reais ele apresenta diversas falhas conforme visto em teste disponibilizado na Internet¹⁰.

Para ser executada pela TV Digital, a classe deve implementar a Interface Xlet do pacote “javax.tv.xlet” e portanto, obrigatoriamente, ter implementados os métodos “InitXlet(), startXlet(), pauseXlet() e DestroyXlet()”. Esses métodos, conjunto com seu contexto, representado pela classe “XletContext” também do pacote “javax.tv.xlet”, são responsáveis por providenciar ao Set-top box da TV o controle da execução do Xlet. Para mais informações sobre o funcionamento dos Xlets, vide capítulo 2.2 JAVA TV.

O protótipo deste trabalho foi realizado com uma classe chamada “XletVideo” que implementa a interface Xlet e um vídeo no formato AVI codificado utilizando o *software* MEncoder disponível no MPlayer¹¹.

A videoaula da disciplina matemática, conteúdo programático função exponencial, foi escolhida porque facilita a explicação da matéria através de exemplos e desenhos de gráficos e proporciona uma abordagem mais detalhada do tema. Ela foi obtida através do endereço http://www.youtube.com/user/nerckie#p/search/0/7_T2JGEqZgg no formato

⁹ Disponível em: <www.twitter.com>.

¹⁰ Disponível em: <<http://www.youtube.com/watch?v=7KRgDheHcPI>>.

¹¹ Disponível em <<http://www.mplayerhq.hu/design7/dload.html>>.

FLV, que não é suportado pelos padrões do Ginga. Para convertê-la para o formato *AVI* foi necessário:

1. Com o MPlayer instalado no computador, acesse o *prompt* de comando do Windows;
2. Navegue até a pasta onde está o MPlayer (Exemplo: C:\Users\Evandro\Desktop\Monografia\MPlayer-p4-svn-33883);
3. Coloque o vídeo a ser codificado na pasta do MPlayer;
4. Digite a linha de comando: `mencoder <NOME_DO_VIDEO>.flv -o <NOVO_NOME_DO_VIDEO>.avi -ovc lavc -lavcopts vcodec=h263:vbitrate=100 -vf scale=352:288 -oac mp3lame`

Observação: No lugar do <NOME_DO_VIDEO>, deve ser digitado o nome do vídeo a ser convertido e no lugar de <NOVO_NOME_DO_VIDEO>, o nome dado ao novo vídeo resultado da conversão do antigo.

Pronto. Agora o novo vídeo será criado dentro da pasta raiz do MPlayer e está compatível com as especificações do Ginga.

A classe *XletVideo* tem o código descrito na seção Anexo.

Para criar o projeto foi necessário configurar suas referências utilizando o “gingaj-emulator.jar” disponível no emulador e a biblioteca “jmf.jar” referente ao *JMF* (para mais informações sobre *JMF* consulte o capítulo 3.4 Ginga-J deste mesmo trabalho).

Para executá-lo através do emulador foi necessário configurar a aplicação no menu "Applications >> Manage Applications..." conforme figura abaixo:

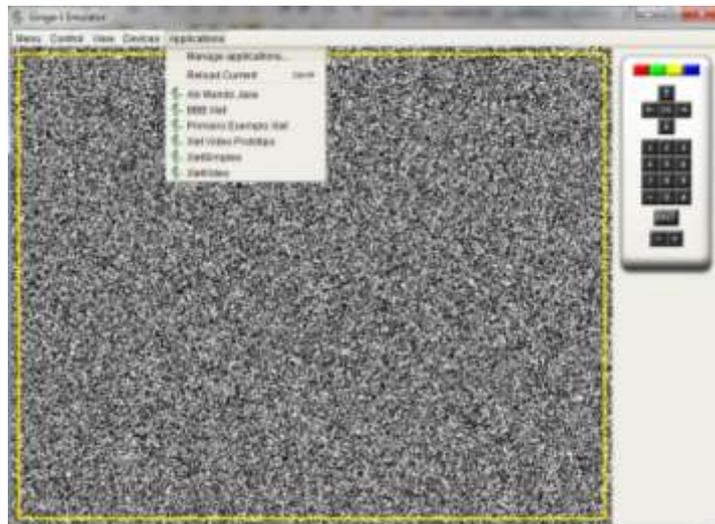


FIGURA 5 – Configurando protótipo no emulador Ginga-J.

Clicar em “Default Group >> New Application”. Após esse passo, um novo item será criado na lista à esquerda abaixo de “Default Group”. Clique com o botão esquerdo sobre esse item e preencha os campos da direita da tela da seguinte forma:

Name – Deve conter o nome da sua aplicação no emulador. (Por exemplo: Prototipo Xlet Video);

Path – Deve conter o caminho físico da sua aplicação no computador até a pasta “bin”. (Por exemplo: C:\workspace\PrototipoXletVideo\bin);

Xlet – Deve conter o caminho para o Xlet a ser executado. (Por exemplo: test.XletVideo).

Agora para executar a aplicação basta clicar em “Applications” e depois no nome escolhido no item **Name** nos passos acima e pronto. O resultado do protótipo rodando com o vídeo pode ser visto na figura 6.

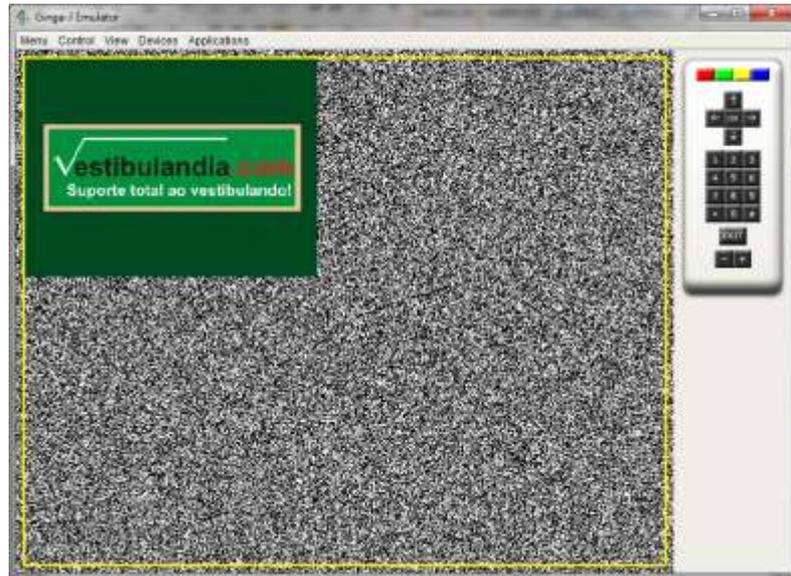


FIGURA 6 – Protótipo rodando no emulador Ginga-J.

CONCLUSÃO

Após a criação do padrão brasileiro de Tv Digital, o Brasil entrou na era da transmissão digital e criação de conteúdo para essa nova realidade. Atualmente esse novo tipo de transmissão é uma realidade em algumas partes do Brasil e está em movimento de expansão para todo o território nacional. Seguindo essa premissa e visando fomentar o desenvolvimento de aplicações para o ambiente brasileiro de Tv Digital Ginga, o laboratório LAViD disponibilizou para a comunidade o *gingaj-emulator*. Com ele foi possível criar o protótipo de um programa utilizando o núcleo procedural do Ginga, o Ginga-J, capaz de exibir um vídeo de matemática e utilizando a Tv como ferramenta de disseminação do conhecimento.

Durante a construção e testes do protótipo, ocorreram tentativas frustradas de configuração do ambiente de desenvolvimento e de uso de algumas classes mais avançadas, através das quais foi possível identificar que o emulador *gingaj-emulator* ainda se encontra incompleto perante a especificação do padrão, o que ainda inviabiliza a construção de aplicações mais robustas. Prova disso foram os resultados dos testes realizados nas aplicações desenvolvidas utilizando o emulador. Tanto o Ginga Hero quanto o Jogo da Velha apresentaram falhas de funcionamento em um ambiente real de Tv Digital conforme provado nos testes realizados pelo projeto OpenITVLab. O

Ginga Hero depois de várias tentativas não foi executado no conversor integrado do televisor Sony, já no Proview ele carregou após alguns minutos, porém se encerrou inesperadamente ao tentar navegar pelos menus e, além disso, ao entrar no jogo a música não se inicia e os comandos do controle não são reconhecidos. No teste realizado com o aplicativo Jogo da Velha, o fluxo de áudio e vídeo do canal travou ao iniciar a aplicação e ao tentar preencher uma casa já preenchida no jogo. O emulador ainda está em fase de desenvolvimento mesmo já tendo sido disponibilizado para a comunidade de programadores. Uma equipe continua recebendo informações de erros por parte dos usuários e trabalhando para resolvê-los através da comunidade Ginga CDN (*Ginga Code Development Network*).

O maior problema encontrado ocorreu na API *xjvax* disponibilizada no emulador Ginga-J pelo LAViD da Universidade Federal da Paraíba como base de desenvolvimento. Entre outras, a classe *Manager*, responsável por criar um *MediaPlayer* capaz de executar um vídeo, está incompleta com todos seus métodos não implementados conforme pode ser visto no código fonte da classe disponível no Anexo II.

Ao tentar instanciar um objeto do tipo *Player* através do método *createPlayer(URL)* da classe supracitada, ocorria erro porque ele sempre retorna nulo uma vez que não estava implementado.

Para resolver esse problema, foi necessário utilizar a API oficial, *javax*, disponibilizada pela Oracle, o que tornou possível construir o protótipo com sucesso e atender à proposta de desenvolvimento, pois esta API não faz parte das APIs previstas para o padrão Ginga-J.

Podemos considerar que a especificação do GINGA-J é robusta, mas seu estado atual de implementação não permite a construção de aplicações procedurais complexas contendo elementos multimídia e muita interação do usuário, uma vez que o comportamento do *software* desenvolvido não pode ser previsto utilizando as ferramentas disponíveis.

REFERÊNCIAS

ANDREÃO, Cristiano Fioresi. Vila Velha: Centro Universitário Vila Velha, 2011. *Conceitos básicos das plataformas Java e J2ME*. Disponível em: <<http://www.devmedia.com.br/post-6484-CONCEITOS-BASICOS-DAS-PLATAFORMAS-JAVA-E-J2ME.html>>. Acesso em: 9 ago. 2011.

COSTA, Bernardo Jacques Delgado. *Transferência de dados via dispositivo móvel utilizando plataforma Java micro edition*. Monografia de Conclusão de Curso. Centro de Ensino Superior de Juiz de Fora. Juiz de Fora, Minas Gerais, 2009.

DIAS, Klessis Lopes; FONTES, Wescley Pimentel. Belém: Universidade Federal do Pará, 2003. *Desenvolvimento de aplicações para dispositivos móveis utilizando a plataforma J2ME*. Disponível em: <http://fabio.bressler.com.br/archives/public/documents/java-klessis_wescley.pdf>. Acesso em: 28 ago. 2011.

DONATO, Maurício. *Como funciona a Tv Digital*. Disponível em: <<http://videos.hsw.uol.com.br/televisao-digital-2-video.htm>>. Acesso em: 29 out. 2011.

DVB PROJECT OFFICE. *Globally Executable Middleware*. 2011. Disponível em: <http://www.dvb.org/technology/fact_sheets/DVB-GEM_Factsheet.pdf>. Acesso em: 9 ago. 2011.

ENCICLOPÉDIA Mirador Internacional, São Paulo: Encyclopaedia Britannica do Brasil, 1998, Volume 14.

GINGACDN. *Usando a máquina virtual*. Disponível em: <http://gingacdn.lavid.ufpb.br/projects/ginga-j/wiki/Usando_a_m%C3%A1quina_virtual>. Acesso em: 01 ago. 2011.

GINGA ORG. 2011. Disponível em: <<http://www.ginga.org.br/pt-br/sobre>>. Acesso em: 9 ago. 2011.

HORA, Lina Cláudia Moreira Garrido. São Paulo: Faculdade Cásper Líbero, 2009. *Talkshow na era da tv digital novo conteúdo, nova recepção, nova programação*. Disponível em: <http://www.facasper.com.br/rep_arquivos/2010/07/22/1279826542.pdf>. Acesso em: 30 ago. 2011.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). *PNAD 2009: rendimento e número de trabalhadores com carteira assinada sobem e desocupação aumenta*. 2009. Disponível em: <http://www.ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=1708>. Acesso em: 9 ago. 2011.

JOKIPELTO, Päivi Aarreniemi. Lappeenranta. Helsinki, 2005. *T-learning model for learning via digital Tv*. Disponível em:

<<http://www.it.lut.fi/eaeie05/proceedings/p21.pdf>>. Acesso em: 28 ago. 2011.

JUNIOR, Marco Aurélio Freesz. *Quiz para TV Digital Interativa: Ambiente completo de execução*. Monografia de Conclusão de Curso. Departamento de Ciência da Computação. Universidade Federal de Juiz de Fora. Juiz de Fora, Minas Gerais, 2010.

MELO, Paulo; RIOS, Evaristo; GUTIERREZ, Regina. 2000. *Tv digital: desafio ou oportunidade?*. Disponível em:

<http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_pt/Galerias/Arquivos/conhecimento/relato/tv_digit.pdf>. Acesso em: 27 ago. 2011.

MORAES, Marcelo Moreira de. Nova Iguaçu, 2009. *A história do surgimento da linguagem Java*. Disponível em:

<http://www.marcelomoraes.com.br/conteudo/marcelo/java/historia_java.pdf>.

Acesso em: 28 ago. 2011.

OLIVEIRA, Angélica; BIANCHINI, David. Campinas: PUC Campinas, 2011.

Desenvolvimento de uma aplicação para TV digital utilizando a plataforma

Ginga J. Disponível em: <[\[campinas.edu.br/pesquisa/ic/pic2011/resumos/2011830_17226_494408304_re_savi.pdf\]\(http://www.puc-campinas.edu.br/pesquisa/ic/pic2011/resumos/2011830_17226_494408304_re_savi.pdf\)>. Acesso em: 29 out. 2011.](http://www.puc-</p></div><div data-bbox=)

ORACLE. *Java ME Technology Overview*. Disponível em:
<<http://www.oracle.com/technetwork/java/javame/java-me-overview-402920.html>>. Acesso em: 9 ago. 2011.

ORACLE. *Java TV*. Disponível em:
<<http://www.oracle.com/technetwork/java/javame/javatv/overview/getstarted/index.html?origref=http://www.oracle.com/technetwork/java/javame/java-me-overview-402920.html>>. Acesso em: 9 ago. 2011.

PÁSCOA, Ana Rita Namora. Portugal, Lamengo, 2006. *A invenção da televisão*. Disponível em:
<http://www.notapositiva.com/trab_estudantes/trab_estudantes/eductecnol/eductecnol_trab/invenctelevisao.htm>. Acesso em: 27 ago. 2011.

PAWLAN, Monica. 2001. *Introduction to Digital TV applications programming*. Disponível em:
<<http://java.sun.com/developer/technicalArticles/javatv/apiintro/index.html>>. Acesso em: 28 ago. 2011.

PINTO, Lady; QUEIROZ NETO, José; SILVA, Kátia. João Pessoa, 2007. *Tecnologias para o desenvolvimento de aplicações educacionais para TV digital*. Disponível em:
<http://www.redenet.edu.br/publicacoes/arquivos/20080212_081200_INFO-078.pdf>. Acesso em: 29 out. 2011.

SANTOS, Joel; RATAMERO, Erick; ARRUDA, João Paulo; DANTAS, Manoel; SANCHEZ, Maria Luiza; SAADE, Débora C. Rio de Janeiro, 2008. Muchaluat. *RummiTV: An Interactive Game for the Brazilian Digital TV System*. Disponível

em: <<http://www.midiacom.uff.br/gtvd/rummitv/downloads/sbrt2008-final.pdf>>. Acesso em: 28 ago. 2011.

SEVERIANO, Alan. *Tv conectada com Internet é a mais nova atração das lojas de eletrônicos. Jornal da Globo, sítio eletrônico na Internet G1, 27 de maio de 2011.* Disponível em: <<http://g1.globo.com/jornal-da-globo/noticia/2011/05/tv-conectada-com-internet-e-mais-nova-atracaao-das-lojas-de-eletronicos.html>>. Acesso em: 28 ago. 2011.

SIERVI, Brian Mazini. *Criação e Distribuição de Conteúdo RSS para a TV Digital na Universidade Federal de Juiz de Fora.* Monografia de Conclusão de Curso. Departamento de Ciência da Computação. Universidade Federal de Juiz de Fora. Juiz de Fora, Minas Gerais, 2010.

SOARES, Luiz Fernando Gomes. Rio de Janeiro: Universidade Católica do Rio de Janeiro, 2009. *TV Interativa se faz com Ginga.* Disponível em: <http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_06_soares.pdf>. Acesso em: 9 ago. 2011.

SOUZA FILHO, Guido Lemos de; LEITE, Luiz Eduardo Cunha; BATISTA, Carlos Eduardo Coelho Freire. João Pessoa: Universidade Federal da Paraíba, [200-?]. *Ginga-J: The Procedural Middleware for the Brazilian Digital TV System.* Disponível em : <<http://bibliotecadigital.sbc.org.br/download.php?paper=625>>. Acesso em: 28 ago. 2011.

SUBGRUPO DE TRABALHO 2 DO COMITÊ DE ATIVIDADES DE PESQUISA E DESENVOLVIMENTO DA AMAZÔNIA (CAPDA). *TV Digital Interativa*. 2004. Disponível em: <http://www.ctpim.org.br/tv_digital.pdf>. Acesso em: 26 ago. 2011.

YOSHIDA, Paulino Shiguelo; FRANCO, Cláudio Bruno Silva de Mendonça. *Universidade Java*. São Paulo: Digerati Books, 2004.

ZANCANARO, Airton; SANTOS, Paloma; TODESCO, José Leomar. São Paulo, 2009. *Ginga-J ou Ginga-NCL: características das linguagens de desenvolvimento de recursos interativos para a TV Digital*. Disponível em: <<http://tvdi.egc.ufsc.br/index.php/observatorio/repositorio-de-artigos/Artigos---Arquitetura-TVD/Ginga-J-ou-GINGA-NCL/>>. Acesso em: 9 ago. 2011.

ANEXO I – CLASSE XLETVIDEO

```
package test;

import java.net.URL;

import javax.media.Manager;
import javax.media.Player;

import xjavadoc.tv.xlet.Xlet;
import xjavadoc.tv.xlet.XletContext;
import xjavadoc.tv.xlet.XletStateChangeException;

import com.sun.dtv.lwuit.Form;

public class XletVideo implements Xlet {

    private XletContext context;
    private Form form;
    private Player player;

    public XletVideo () {
        form = new Form();
    }

    @Override
    public void destroyXlet(boolean arg0) throws XletStateChangeException
    {
        context.notifyDestroyed();
    }

    @Override
    public void initXlet(XletContext xletContext) throws
XletStateChangeException {
        context = xletContext;
    }

    @Override
    public void pauseXlet() {
```

```
}  
  
@Override  
public void startXlet() throws XletStateChangeException {  
    try {  
        //Cria o objeto vídeo para ser rodado pela aplicação  
        URL url = new URL ("file:///C:/aulaMatematica.avi");  
  
        player = Manager.createPlayer(url);  
  
        com.sun.dtv.lwuit.MediaComponent mediaComponent = new  
com.sun.dtv.lwuit.MediaComponent(player);  
  
        //Adiciona o componente de vídeo à tela  
        form.addComponent(mediaComponent);  
  
        //Inicia a execução do vídeo  
        mediaComponent.startMedia();  
  
        // Mostra o form na tela  
        form.show();  
    } catch(Exception e) {  
        e.printStackTrace();  
    }  
}  
}
```

ANEXO II – CLASSE MANAGER

```

package xjavadoc.media;

public final class Manager extends Object {

    public static final String UNKNOWN_CONTENT_NAME = "Unknown
content";

    public static Player createPlayer(java.net.URL sourceURL)
throws java.io.IOException, NoPlayerException {
        return null;
    }

    public static Player createPlayer(MediaLocator sourceLocator)
throws java.io.IOException, NoPlayerException {
        return null;
    }

    public static Player
createPlayer(xjavadoc.media.protocol.DataSource source)
throws java.io.IOException, NoPlayerException {
        return null;
    }

    public static xjavadoc.media.protocol.DataSource
createDataSource(java.net.URL sourceURL)
throws java.io.IOException, NoDataSourceException {
        return null;
    }

    public static xjavadoc.media.protocol.DataSource
createDataSource(MediaLocator sourceLocator)
throws java.io.IOException, NoDataSourceException {
        return null;
    }

    public static TimeBase getSystemTimeBase() {
        return null;
    }
}

```

```
    public static java.util.Vector getHandlerClassList (String
contentName) {
        return null;
    }
}
```