

Java TV: Visão Geral

Ricardo da Silva Ogliari
Juliana Rosa Santos

Introdução

Depois que o governo brasileiro definiu o padrão de TV Digital no Brasil, o assunto, de certa forma, tomou proporções elevadas, sendo difundido em diversas mídias nacionais. Porém, a especificação da linguagem Java, designada especialmente para fornecer uma plataforma de desenvolvimento para este tipo de equipamento, chamada de Java TV, já estava disponível antes mesmo do anúncio oficial do governo federal.

Da mesma forma que acontece com diversas especificações da linguagem, também foi criado um emulador para simular uma TV Digital em um *desktop*, chamado de XLetView. Esse emulador é baseado na middleware MHP e é Open Source. Ele fornece uma maneira fácil e rápida de testar as aplicações, chamadas de Xlets.

O presente artigo objetiva apresentar de forma clara e rápida, uma visão rápida sobre a Java TV, demonstrando trechos de código para uma estrutura de aplicativo. Em seguida, fornece um breve roteiro de como o leitor pode compilar e emular seus códigos para TV Digitais na ferramenta XLetView.

TV Digital

A TV Digital ganhou projeção nacional nos últimos meses com o anúncio do padrão a ser adotado no Brasil. Sua adoção se dará gradativamente, porém, pode-se adiantar que seus benefícios serão grandes. Alguns autores comparam a mudança na transmissão da TV de analógico para digital, com a mudança da TV preto e branco para a TV colorida. Apesar da produção dos programas de TV já serem feitos de forma digital, o sinal ainda é transmitido de forma analógica. Além de fornecer melhor qualidade de som e imagem e possibilidade de transmitir diversos programas em um mesmo canal, seus sinais poderão ser capturados por diversos tipos de equipamentos, como em um telefone celular. E, quem sabe, em um futuro próximo, poderemos assistir ao jogo de nosso time preferido na tela de nosso *smartphone*, no relógio de pulso, seja em casa, ou dentro do carro, ônibus e assim por diante.

Além destas mudanças, adiciona-se a questão da interatividade, talvez a maior mudança no já costumeiro ambiente passivo do qual o telespectador assiste a seu programa de TV. Ao mesmo tempo em que assiste a seu jogo preferido, o telespectador poderá receber informações sobre cada jogador, notícias do campeonato, jogos, além de poder usufruir de uma nova forma de comércio eletrônico que já é chamada de *t-commerce*, entre outras inúmeras outras formas de interatividade. Estudos apontam para uma mudança de comportamento, onde o telespectador poderá vir a se transformar em um usuário, assim como funciona no mundo da Internet.

Além de tudo isso, a TV pode se tornar um elemento de extremo interesse para o comércio de grande porte, simplesmente pelo fato da maioria dos domicílios possuírem um televisor ou mais, e, as poucas pessoas que não usufruem deste equipamento, podem encontrar informações vindas pela TV em restaurantes, bares, etc.

Fica claro que este ramo não deve ser ignorado. Seu potencial é grande, e se a idéia da TV Digital realmente vingar no Brasil, muito conteúdo digital, desde entretenimento até negócios vai ser demandado, cabendo a nós, desenvolvedores, suprir estas tendências de mercado.

Java TV

Talvez seja um pouco de exagero, mas uma das coisas mais fascinantes em Java é sua alta granularidade, ou seja, se você quer programar para um automóvel, existe a Java Car, se deseja que um anel com capacidades de processamento execute um programa específico, dê uma olhada em JavaRing, além de inúmeros outros casos. Dessa forma, não seria diferente com TV Digital, onde, existe uma completa API, chamada de Java TV, que é um pacote opcional para a plataforma J2ME, mais especificamente para o configuração CDC (Connected Device Configuration), que fornece a você, 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

O objetivo deste artigo não é entrar em detalhes sobre a API, sendo assim, o leitor deve reler o último item listado acima, que diz que a Java TV fornece um gerenciamento do ciclo de vida das aplicações. Para aqueles que já conhecem o funcionamento de um *applet* ou de uma *MIDlet*, já sabem do que se trata, para quem não é familiarizado com estes termos, entenda este conceito como os estados possíveis de uma aplicação, veja a Figura 1. Os programas em Java TV são chamados de *XLet*.

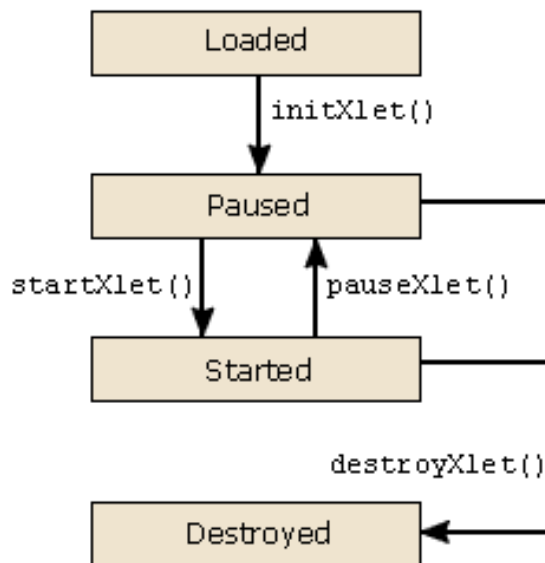


Figura 1 – Estados de uma Xlet. **Fonte:** Interactive Tv Web

Uma vez que a classe é carregada no aparelho ela está no estado *Loaded* (só entra neste estado uma única vez), a partir deste momento, a aplicação pode ser executada, entrando no estado *Started*, podendo, posteriormente, ser pausada (*Paused*, estado onde a aplicação Xlet é iniciada) ou destruída (*Destroyed*, onde termina o ciclo de vida). Semelhante ao que ocorre com *applets* e *MIDlets*, as *Xlets* também fornecem métodos para um completo gerenciamento do ciclo de vida de uma aplicação.

Veja a Listagem 1.

```
public interface Xlet {
    public void initXlet(XletContext context) throws
XletStateChangeException;

    public void startXlet() throws XletStateChangeException;

    public void pauseXlet();
    public void destroyXlet(boolean unconditional) throws
XletStateChangeException;
}
```

Listagem 1 – Construtor da classe JTable

Em relação à interface gráfica, a Java TV utiliza componentes AWT definidos pelo perfil J2ME onde está sendo executado, fornecendo um container para comunicação com o componente pai.

Já a parte de controle de mídia a especificação inclui a Java Media Framework 1.0 (JMF).

A especificação da API do Java TV é formada pelos seguintes pacotes:

- `javax.tv.graphics`: habilita simples composições e fornece um container para construção de interfaces com componentes AWT.
- `javax.tv.locator`: fornece acesso aos dados e recursos endereçados ao receptor de TV digital.
- `javax.tv.media`: define uma controle e eventos para gerenciamento de conteúdos visuais e de áudio.
- `javax.tv.net`: fornece acesso aos datagramas IP transmitidos pelo stream de broadcast.
- `javax.tv.service`: fornece acesso ao banco de dados do service information (SI) e APIs para seus sub-pacotes.
- `javax.tv.util`: suporta a criação e gerenciamento de eventos de tempo.
- `javax.tv.xlet`: fornece os métodos para gerenciamento do ciclo de vida da Xlet controlados pelo gerenciador de programas.

Sugere a leitura da documentação da Java API para um melhor esclarecimento sobre os pacotes e suas classes.

Minha primeira Xlet

Para compilar nosso primeiro exemplo, primeiro deve-se baixar a ferramenta XletView, que se encontra neste endereço: http://sourceforge.net/project/showfiles.php?group_id=81591&package_id=83543&release_id=243876. Depois de baixar o arquivo zip, é necessário que o mesmo seja deszipado em qualquer diretório, com exceção do diretório onde encontra-se as classes. Dentro desta pasta, é encontrado um arquivo chamado `xletview.jar`, onde encontramos todas as classes que vamos precisar para construir nossa primeira Xlet, portanto, adicione esta biblioteca ao seu `classpath`.

Posteriormente, copie o código encontrado **aqui**. Ele simula uma pequena urna eletrônica que seria executada através de uma TV Digital. Alguns pontos importantes do código merecem destaque. O primeiro ponto são as classes importadas, perceba que a Java TV também utiliza um pacote `org.havi.ui` para construção da interface do aplicativo. Veja a Listagem 2.

```
import javax.tv.xlet.Xlet;
import javax.tv.xlet.XletStateChangeException;
import javax.tv.xlet.XletContext;

import org.havi.ui.HScene;
import org.havi.ui.HSceneFactory;
import org.havi.ui.HScreen;
```

Listagem 2 – Imports necessários.

O segundo ponto importante no código é a construção da cena que será mostrada no display do aparelho. Através da classe *HSceneFactory* e de seu método *getFullScreenScene* é obtida uma instância da classe *HScene*, onde serão adicionados os componentes AWT, perceba que logo abaixo de sua construção é configurado alguns parâmetros, como seu tamanho, localização e layout.

```
HSceneFactory hsceneFactory = HSceneFactory.getInstance();
scene = hsceneFactory.getFullScreenScene(
    HScreen.getDefaultHScreen().getDefaultHGraphicsDevice());
scene.setSize(200, 510);
scene.setLocation(100, 100);
scene.setLayout(null);
```

Listagem 3 – Construção da cena.

Deve-se levar em consideração que um Xlet só usar componentes AWT puros, dessa forma, alguns desenvolvedores tem preferido desenhar seus próprios componentes. A API JavaTV não possui todos os componentes. Por exemplo, um *check-box* ainda não é disponível. Neste código, foram usados 3 *buttons*, 2 *lists*, um *scene* e 2 *panels*. Um dos *panels* é usado para apresentar a listas e mostrar o resultado da votação.

```
private XletContext context;
private HScene scene;
private Panel panel = null;
private Panel panelVotos = null;
private List slist = null;
private List slistVotos = null;
private Button button = null;
private Button buttonResultado = null;
private Button buttonLista = null;
```

Listagem 4 – Componentes

Para testar este aplicativo basta executar o *xletview.jar* que se encontra no diretório raiz do arquivo gerado pelo .zip. Para usuário do Windows, basta entrar nesta pasta via DOS e passar a seguinte linha de comando:

```
java -jar xletview.jar
```

O *prompt* de comando deve ficar conforme a Figura 2. Em seguida, uma tela semelhante à Figura 3 deverá aparecer.

```

c:\ Prompt de comando
C:\Documents and Settings>cd juliana
C:\Documents and Settings\Juliana>cd desktop
C:\Documents and Settings\Juliana\Desktop>cd xlet
C:\Documents and Settings\Juliana\Desktop\XLET>cd xletview-0.3.6
C:\Documents and Settings\Juliana\Desktop\XLET\xletview-0.3.6>java -jar xletview.jar
*****
XleTView, Copyright (C) 2003 - 2004 Martin Sveden
XleTView comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute
it under certain conditions;
see license document for details.
*****
setting properties...
free/used/total: 2331 K / 4324 K / 6656 K
running gc...
after gc...
free/used/total: 3122 K / 3533 K / 6656 K
[XleTView]-INFO->loading Xlet... [Uotar]
[XleTView]-INFO->XLET started... [Uotar]
C:\Documents and Settings\Juliana\Desktop\XLET\xletview-0.3.6>

```

Figura 2 – Prompt de comando DOS

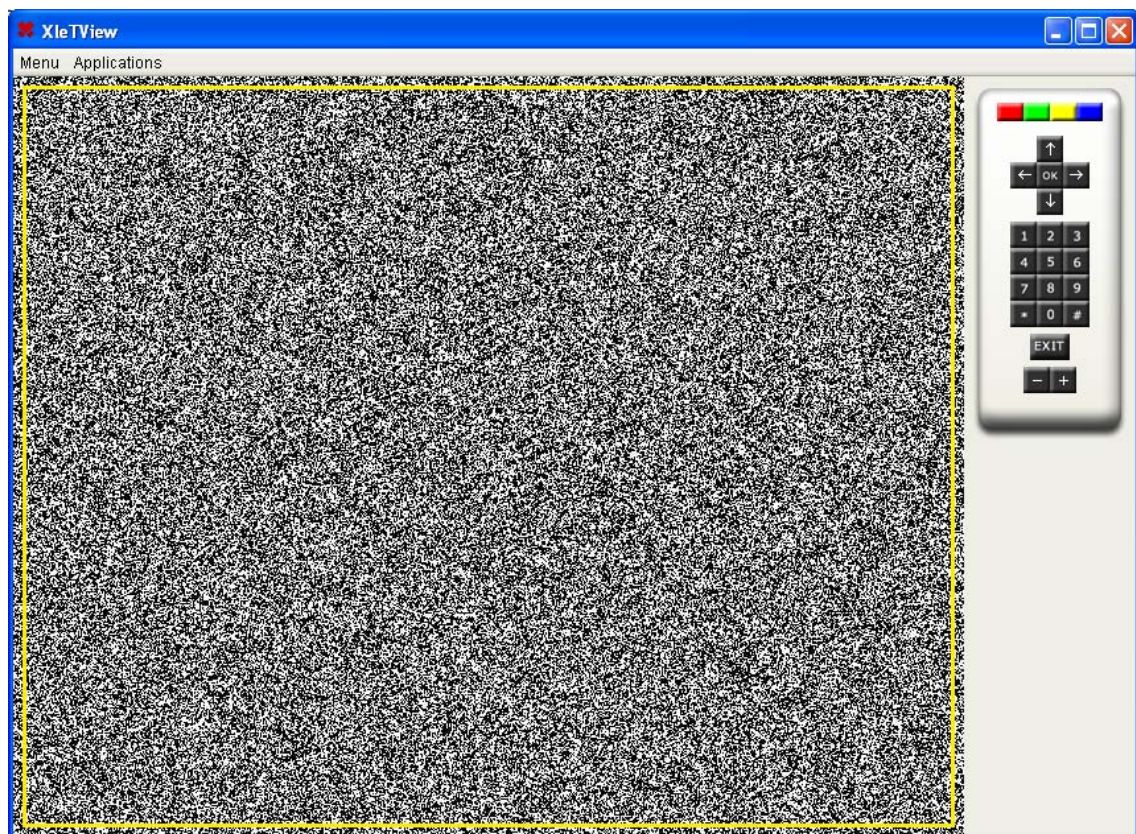


Figura 3– XLetView.

Para executar nosso exemplo, entre em *Applications->Manage Applications*, uma tela semelhante à Figura 4 deverá aparecer. Selecionando *New Application* mais um item deve aparecer na lista da esquerda, chamada de *new app1*, selecione este item, suas propriedades aparecem na parte direita. O

nome fica a sua escolha, em *Path* indique o caminho onde está a classe (.class) que você compilou, pode ser a do exemplo anterior ou de outra qualquer, por fim, em *Xlet* indique qual classe identificará a *Xlet* propriamente dita. O próximo passo é salvar estas mudanças, clique em *Save & Close*, na parte inferior da mesma tela. Voltando a tela principal do emulador, vá a *Applications* e selecione a que acabou de criar. Seu programa deverá executar na tela de sua “TV Digital”. Se optar por configurar como sua *Xlet* o exemplo citado anteriormente, você deverá ver uma tela semelhante à Figura 5.

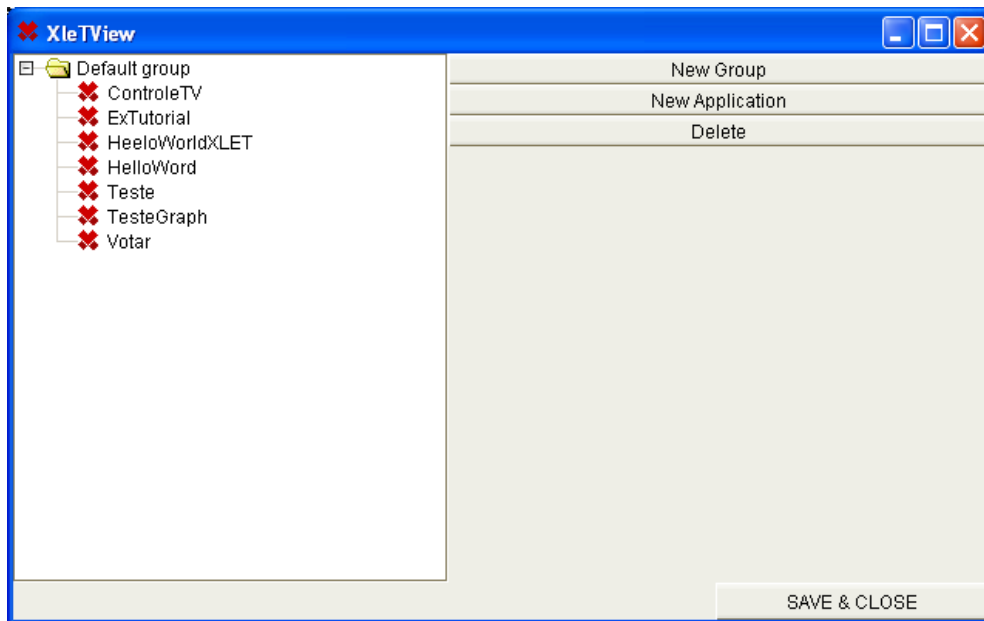


Figura 4 – Tela de gerenciamento de aplicações.

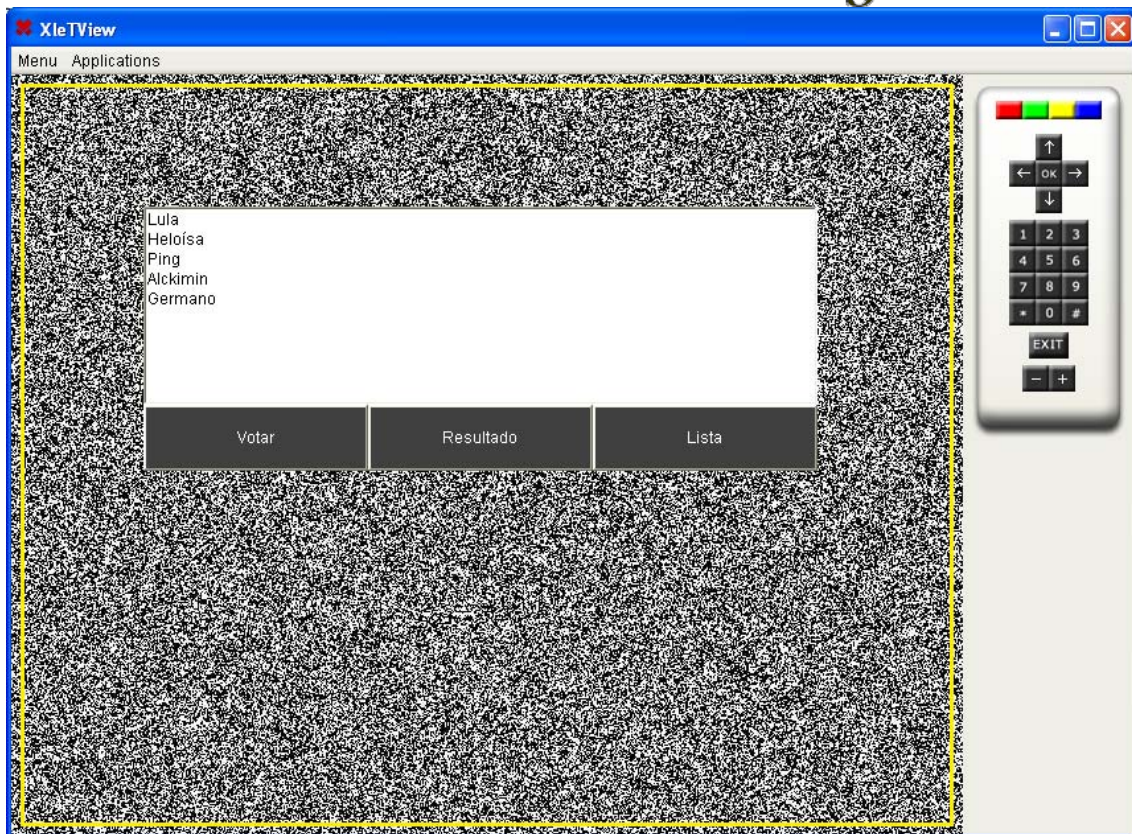


Figura 5 – Programa de votos sendo executado no emulador.

Conclusão

A TV Digital no Brasil ainda encontra-se em fase embrionária, diferente de outros países onde já existem padrões definidos há alguns anos, porém, seu potencial de mercado não deve ser ignorado por programadores e empresas da área. Para tanto, existe a Java TV, uma API completa para programação de qualquer sistema que seja direcionado a este novo tipo de equipamento. Cabe ao leitor se aperfeiçoar neste novo nicho de mercado e ter benefícios quando a TV Digital realmente começar a decolar no Brasil.

Referências Bibliográficas

INTERACTIVE TV WEB. An introduction to Xlets. Disponível em <<http://www.interactivetvweb.org/tutorial/mhp/xletintro.shtml>>.

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/~tg/2003-2/jal-proposta.do>.

PAWLAN, Mônica. Introduction to Digital TV Applications Programming. 2001. Disponível em <<http://java.sun.com/developer/technicalArticles/javatv/apiintro/>>.

Sobre os autores

Ricardo da Silva Ogliari é formando em Ciência da Computação pela Universidade de Passo Fundo – RS. Desenvolveu sistemas mobile para empresas do Rio Grande do Sul, atualmente trabalha como desenvolvedor de aplicações direcionadas a telefones celulares na Kwead.com de São Paulo. Também, foi palestrante dos eventos JustJava 2005 e EJES (Encontro Java do Espírito Santo) 2005, além de ministrar alguns cursos sobre a tecnologia J2ME.

Juliana Rosa Santos é formando em Sistemas de Informação pela Pontifícia Universidade Católica de Minas Gerais. Faz parte da equipe de projeto de pesquisa do Centro de Sensoriamento Remoto da Universidade Federal de Minas Gerais, onde desenvolve sistemas em Java pra modelagem de cenários para Amazônia.