

# Infra-Estrutura para Recepção de TV Interativa Baseada em *SetTopBox* para o Projeto I2TV

Proposta de apresentação de trabalho na área de Vídeo Digital para o 4º Workshop RNP2

Leonardo A. Ribeiro                      Marcos F. Caetano                      Alexandre Schulter  
leoar@lisha.ufsc.br    caetano@lisha.ufsc.br    schulter@inf.ufsc.br  
Valdecir Becker                              Calos Montez  
valdecir@nurcad.ufsc.br    montez@das.ufsc.br  
Edison Melo                                  Antônio A. M. Frohlich  
melo@npd.ufsc.br    guto@lisha.fsc.br

Laboratório de Integração Software e Hardware - LISHA  
Núcleo de Redes de Alta Velocidade e Computação de Alto Desempenho - NURCAD  
Centro Tecnológico - Universidade Federal de Santa Catarina - UFSC/CTC  
Caixa Postal 476 - 88040-900 - Florianópolis - SC

## Resumo

As soluções encontradas atualmente para plataformas receptoras de TV digital, chamados *SetTopBoxes*, são caracterizadas por serem produtos proprietários, de soluções fechadas. O presente artigo apresenta os trabalhos iniciais realizados com o intuito de desenvolver um *SetTopBox* utilizando soluções abertas e que seja portátil entre diferentes arquiteturas de *hardware* e fontes de conteúdo. O conjunto de tarefas foi dividido basicamente em duas frentes: a escolha, preparação e aprimoramento da plataforma de *Hardware* e sistema operacional responsável por receber e decodificar o *streaming* de dados de TV Digital, e o ambiente de software que provê o controle semântico sobre as informações de mídia e aplicações desenvolvidas especialmente para possibilitar interatividade.

## 1 Introdução

Sistemas de TV Digital requerem uma plataforma receptora para tratamento do conteúdo digital compreendendo componentes de *hardware* dedicados e uma ampla e flexível camada de *software* para prover todos os novos recursos que este sistema oferece, por exemplo, aplicações interativas multiplexadas junto com fluxo de mídia. Os dispositivos que implementam esta plataforma são comumente chamados de *SetTopBoxes*, sendo comercializados em países que utilizam TV Digital, como Estados Unidos, Japão e diversos países europeus. O Brasil, por sua parte, vem avaliando tecnologias, definindo políticas e executando projetos pilotos, como o I2TV, projeto que fomentou os experimentos que conduziram a este artigo.

O projeto I2TV almeja desenvolver tecnologias e ferramentas de suporte à TV Interativa no ambiente Internet2 [3]. Este artigo descreve o estado atual do projeto nos trabalhos em infraestrutura de recepção baseada em *SetTopBox*, o qual vem sendo desenvolvido no LISHA/UFSC com enfoque na utilização e criação de tecnologias “abertas”. Esta decisão de projeto conduziu à escolha do sistema operacional LINUX e da API JAVATV como componentes fundamentais da plataforma ao lado do padrão *MPG2 TS* que é consenso nas discussões sobre televisão digital.

A próxima seção descreve a infra-estrutura de hardware e sistema operacional que compõem a subcamada receptora de TV digital. Em seguida são descritos o conjunto de APIs JavaTV que desempenham o papel de *middleware* entre as aplicações e a subcamada de hardware e as adaptações realizadas na mesma para adequá-la aos objetivos do trabalho. Finalmente a última seção descreve o protótipo inicial desenvolvido e os trabalhos futuros a serem realizados. Este artigo visa principalmente registrar a metodologia utilizada nesse processo, bem como as dificuldades e soluções apresentadas para as mesmas.

## 2 Infra-Estrutura

Partindo-se da necessidade de se criar uma infra-estrutura de recepção de TV Interativa, baseada na recepção de *streams* de video MPG2, definiu-se uma arquitetura de *SetTopBox*. Dentre todos modelos pesquisados algumas características básicas foram observadas, as quais apresentam-se na primeira camada da *Figura 1*

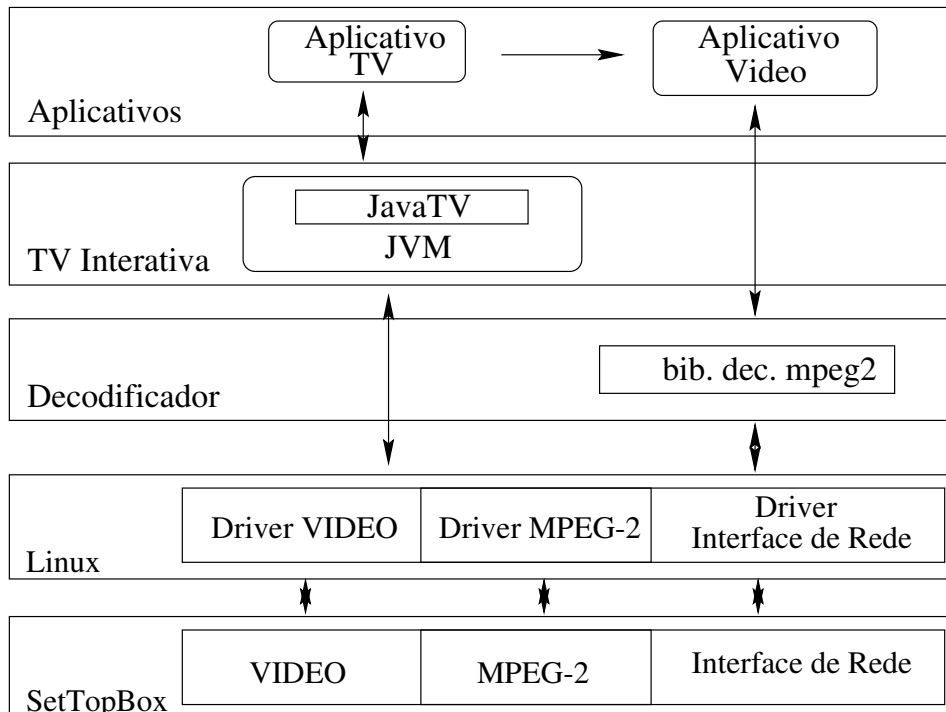


Figura 1: Disponibilização da Plataforma *SetTopBox*

Examinando o diagrama da *Figura 1*, destaca-se seus aspectos mais pertinentes:

- A primeira camada corresponde ao *hardware*, ainda destacamos a *interface* de *video*, Codificador e/ou Decodificador MPG2 e *interface* de rede.
- A segunda camada no diagrama representa o sistema operacional *Linux*, destacando-se os *drivers* necessários para realizar a conexão entre os aplicativos e as *interfaces de hardware*.
- A quinta camada refere-se aos aplicativos para TV digital. Estes realizam interação com a quarta e terceira camadas, seja para obter ponto de acesso para serviços através da *API JavaTV* ou acesso as bibliotecas de decodificação *MPEG2*.

A necessidade dos recursos necessários para a execução de um aplicativo é determinada pela aplicação que irá executar na plataforma, em nosso caso o *aplicativo TV*. É fornecido um conjunto de *APIs JavaTV* a qual irá utilizar dos recursos fornecidos pelo *kernel* do *linux*. São necessários uma interface de rede, um decodificador de *MPG2* e uma placa de *video* juntamente com seus respectivos *drivers* para fornecer a aplicação a capacidade de receber, decodificar e exibir os *streams* de vídeo.

### 3 A plataforma JavaTV

A escolha da utilização do conjunto de APIs JavaTV [1], que constituem a extensão da plataforma Java para o desenvolvimento de conteúdo destinado à televisão interativa, foi motivada principalmente pela existência de uma Implementação de Referência (IR) gratuita e de código fonte aberto disponibilizada pela Sun no final do ano de 2000. Outro motivo foram os trabalhos anteriores realizados pelos parceiros do projeto I2TV sediados em Natal que utilizavam esta mesma IR, o que garante um conhecimento prévio do assunto para o projeto.

Sob ponto de vista da adoção do JavaTV pelo mercado, pode-se considerar que isto vem ocorrendo de maneira indireta através do padrão Multimedia Home Platform (MHP), produzido pelo consórcio europeu *Digital Video Broadcasting* (DVB), o mesmo que produz especificações para codificação de canal de transmissão. Além da Europa, o MHP vem sendo difundido em países que utilizam padrões de transmissão diferentes do DVB, e caminha para tornar-se um padrão de fato no desenvolvimento de conteúdo para TV digital. O MHP suporta boa parte das APIs do JavaTV, que é considerado um subconjunto do mesmo. Diversos pontos em comum entre o MHP e o JavaTV, como a utilização do *Java Media Framework* (JMF), o gerenciamento dos serviços e controle do ciclo de vida de aplicações, possibilitam o desenvolvimento de conteúdo para JavaTV que requeira pouca modificação para ser compatível com plataformas MHP.

Na IR do JavaTV, o *setTopBox*, como fonte de informações pertinentes ao conteúdo (e não o conteúdo em si), é abstraído pelas classes do pacote *com.sun.tv.receiver*. As informações sobre as fontes transmissoras, serviços, programação, detalhes de componentes são obtidas de um arquivo em formato *Extended Markup Language* (XML). Este arquivo XML é interpretado pela classe *ReceiverFile*, durante a inicialização da IR.

O principal problema enfrentado nos testes com o IR do JavaTV foi a falta de suporte para decodificação o MPEG2 com multiplexação TS (Transport Stream) pelo JMF. Buscando a resolução deste problema, duas abordagens poderiam ser utilizadas. A primeira é a utilização do *Java Native Interface* (JNI) para acesso às bibliotecas ou mesmo acesso direto aos *drivers* de uma placa decodificadora MPEG2. A vantagem desta abordagem é a grande flexibilidade para acesso aos recursos do sistema providos pelo sistema operacional e portabilidade entre diferentes plataformas. A desvantagem é a dificuldade de implementação. A outra abordagem é a utilização de um aplicativo externo ao controle do ambiente Java. A principal vantagem desta solução é a facilidade de implementação. Esta solução possui como desvantagem a perda do controle sobre o fluxo de mídia e da janela de reprodução, o que não ocorre na primeira abordagem, onde é possível utilizar o código nativo Java apenas na implementação do *front-end* gráfico.

Como o objetivo dos trabalhos, neste estágio do projeto, é o desenvolvimento de um protótipo, obteve-se pela utilização de um aplicativo externo, que nos permitiu dispor mais rapidamente de uma infra-estrutura para testes de aplicações de TV interativa. Foi escolhido *Video Lan Client* (VLC) desenvolvido em *Ecole Centrale Paris*. A motivação para esta escolha, além do código aberto, foi o suporte a fonte de dados em *multicast* e o trabalho já desenvolvido pelos

parceiros do projeto na geração de conteúdo usando o *Video Lan Server* (VLS) [4]. Para contornar parte dos problemas citados anteriormente em utilizar aplicativos externos serão realizadas alterações no código do VLC para usa-lo como *backend* do JavaTV. Especificamente será retirado do VLC seu tratamento aos eventos gerados pelo teclado para que o mesmo possa receber comando simples de sua entrada padrão.

## 4 Protótipo

A fim de demonstrar os conceitos e técnicas descritas anteriormente, um protótipo da plataforma foi implementado. Este protótipo faz uso de duas arquiteturas peculiares como *SetTopBox*: o TIVO-2 [2] da *Tivo* baseado em MIPS e o XBOX da *Microsoft* baseado na INTEL ARCHITECTURE 32. Este último, que originalmente trata-se de um video game, passou por adaptações de hardware para que pudesse desempenhar o papel de um *SetTopBox*. O sistema operacional LINUX com *kernel 2.4* foi instalado em ambas as plataformas de hardware, definindo assim a segunda camada da arquitetura descrita na seção 2.

A quarta e a terceira camada da arquitetura, são constituídas pelo *JAVA2 SDK* da Sun Microsystems, pelo *JavaTV 1.0* e pelas bibliotecas nativas do VLC, respectivamente. Estas camadas da arquitetura mostraram ser peças fundamentais na definição do desempenho geral do sistema, pois grande parte destes componentes de software foram desenvolvido para serem utilizado em computadores pessoais com recursos abundantes, o que não é o caso dos *SetTopBoxes* escolhidos.

A camada de aplicação desenvolvida é baseada em aplicações Java chamadas *Xlets*, que são análogas aos *applets*, mas com escopo de execução em receptores de TV digital. Como primeiro experimento, foi implementado um Guia de Programação Eletrônico, que é atualizado sempre que ocorre alguma mudança na grade de programação. Modificações foram necessárias no ambiente JavaTV para integra-lo com a janela de reprodução de mídia.

## 5 Considerações finais

Com trabalhos descritos neste artigo, possuímos o protótipo de uma plataforma receptora de TV interativa. Mais testes de performance e avaliações à respeito sobre o consumo de recursos ainda precisam ser realizados para aprimoramentos da plataforma descrita, mas de um modo geral dispomos de um ambiente para o desenvolvimento e testes de conteúdo de TV interativa. Dentre os trabalhos futuros, está a implementação de componentes da especificação MHP à partir da IR do JavaTV, desenvolvimento de aplicações Java mais complexas e com maior sincronização com o fluxo MPEG2.

## Referências

- [1] CALDER B.; COURTNEY J.; et al. **JavaTV™ API Technical Overview**, v.1.0. Disponível em <http://www.java.sun.com/products/javatv/>. Acesso em 18 jan. 2003.
- [2] Disponível em <http://www.tivo.com>. Acesso em 31 mar 2003.
- [3] Disponível em <http://www.i2tv.ufsc.br>. Acesso em 20 fev 2003.
- [4] Disponível em <http://www.videolan.org>. Acesso em 20 fev 2003.