Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

RAFAEL SEBASTIÃO BERNINI

Desenvolvimento de Aplicações Sticker para TV Digital

Orientador: **Dr. Almir Rogério Camolesi**

Assis

Dezembro de 2013

FICHA CATALOGRÁFICA

BERNINI, Rafael Sebastião.

Desenvolvimento de Aplicações Sticker para TV Digital. Rafael Sebastião Bernini

Fundação Educacional do Município de Assis – FEMA – Assis, 2013.
Páginas 47.

Orientador: Dr. Almir Rogério Camolesi
Projeto de Iniciação Científica - Instituto Municipal de Ensino Superior de Assis – IMESA.

Resumo

Os padrões de referência para sistemas de TV digital e middleware Brasileira são Ginga e Astro Dev Net da TOTVS. Tais estruturas servem para a autoria de programas para a TV digital interativa utilizando linguagem de declaração NCL (Nested Context Language), scripts, protótipos, linguagem LUA e Java ME (Java Micro Edition). Neste trabalho, ilustra-se o uso dessas linguagens que dão suporte a diversas aplicações como jogos, software de mídia áudio e visual, entretenimento com redes sociais, navegação web, entre uma demanda gigantesca de imenso poder que é permitido de se criar com essas linguagens e suporte a TV digital.

Sumário

Tabelas

Imagens

Código Fonte

Listagem de Programas

1 Introdução

Existem mudanças da antiga TV Analógica para a TV Digital se faz sentir, pela imensa qualidade de imagem e som, guiado por um sistema de transmissão não guiado, que é o que acontece com a TV digital terrestre e analógica e digital, o canal de transmissão introduz diversos tipo de interferência e ruídos no sinal original, limitando quase toda a capacidade do sistema.

O ruído é aleatório e está quase sempre presente em todo o aspecto de frequência e não pode evitado, em sua transmissão analógica, ele provoca queda na qualidade do sinal recebido, causando o aparecimento de “chuviscos” na imagem. A queda de qualidade do sinal recebido depende da relação entre as potência do sinal e do ruído. À medida que a relação diminui, isso pode acontecer pela atenuação do sinal quando nos afastamos da fonte, diminui também a qualidade do sinal recebido. Nos sistemas digitais, o ruído pode modifica a um nível, aumentando a probabilidade de erro de bit. Para evitar o problema, todos os padrões de transmissão para a TV digital utilizam código corretores de erro. Se a taxa de erros estiver abaixo de um limiar, o código corretor é capaz de corrigir todos os serros e introduzidos pelo canal e não há queda da qualidade de imagem, o sistema não é capaz de corrigir o código e pode até introduzir erros, aos invés de corrigir. Por isso, é usual se dizer que a TV digital teremos um sinal perfeito, sem nenhum “chuvisco”, ou nenhum sinal. Como a relação do ruído decai com a distância do transmissor, caso o sistema não esteja bem dimensionado ou direcionado, pode haver problemas de cobertura, com a introdução áreas de sombra.

A deterioração de um sinal recebido pode variar de seus múltiplos pontos percorrido até seu destino final, partindo de um ponto forte até os seus limites presentes. Cada percurso apresenta uma atenuação e atraso diferentes dos demais, fazendo com que o sinal recebido seja formado pela sobreposição dos vários provenientes dos diferentes caminhos.

A TV digital de múltiplos percursos podem produzir uma interferência entre os símbolos, isto é, a sobreposição entre os bits transmitidos, se o intervalo de duração

de um símbolo transmitido é muito pequeno, com como o intervalo entre símbolos, a ponto da diferença de retardo entre múltiplos pontos através do percurso.

O impacto da TV digital é muito mais significativo do que uma simples troca de um sistema de transmissão analógico para digital, e muito mais do que uma melhora da qualidade de imagem e de som. Mais do que isso, um sistema de TV digital permite um nível de flexibilidade inatingível com a difusão analógica. Um componente importante dessa flexibilidade é a habilidade de expandir as funções do sistema por aplicações construída sobre a base de um sistema padrão de referência.

Sendo assim uma das características mais importantes da TV digital é a interação de uma capacidade computacional no dispositivo receptor, permitindo o surgimento de uma grande quantidade de serviços e desenvolvedores, com sistemas e aplicativos que facilita o uso e a interatividade do usuário com maior segurança e facilidade ao uso diário.

1.1 Estrutura do trabalho

Este trabalho encontra-se organizado da seguinte forma: inicialmente, no Capítulo 1, foi apresentada uma visão geral do trabalho. Na sequencia, o Capitulo 2 apresenta uma breve revisão bibliográfica dos conceitos de TV-Digital. No Capítulo 3 é apresentado os conceitos da linguagem para desenvolvimento de aplicações de TV-Digital NCL. Por fim, no capítulo 4 são apresentadas algumas conclusões e trabalhos futuros.

2 Conceitos de TV Digital Interativa

A TV digital se resume a um típico sistema de cliente/servidor, ao qual o servidor compõe um sistema de rádio difusão ou de um servidor de conteúdo, já a base do cliente que seria o usuário telespectador.

Áudio e Vídeo são os principais componentes de um programa podendo ser ao vivo ou gravado mais ambos dependendo de uma câmera e um áudio principal, que são provem idos de um servidor de vídeo e de dados adicionais, incluindo o aplicativo que define o relacionamento entre vários objetos de mídia. Os dados adicionais podem vir encapsulados no formato IP ou em outros formatos. O vídeo e áudio são entregues e decodificadores digitais que são os responsáveis pela geração dos fluxos de áudio e vídeo ambos comprimidos. Esses fluxos e mais os fluxos de dados são então multiplexados em um único sinal, denominado fluxo de transporte. Esse sinal é então modulado para um canal de frequência e transmitidos, no caso de um sistema terrestre ou ar.

2.1 Stikers

Stickers são aplicativos que podem ser baixados para o receptor via canal de interatividade ou através do sinal de TV Digital transmitido por emissoras que suportem a aplicação.

Com os Stickers é possível acessar diversas funcionalidades e serviços, assim como conteúdo que são complementares à programação.

2.2 Codificação do Áudio

Em geral um sinal digital leva consigo informações redundantes, quando se elimina essas redundâncias, conseguimos reduzir reduzir absurdamente essa quantidade de bits gerados a mais. Na TV digital a compressão percentual sem perdas são empregadas para o sinal de áudio, levando em conta o modelo de audição humana, que por fim leva um áudio de alta qualidade e com baixa taxa de bits gerada.

Os padrões americanos de TV digital tem um padrão codificação AC3/ATSC[ATSC AC3 2005]. A codificação, proprietária da empresa Dolby, utiliza 6 canais de áudio, sendo 5 com alta qualidade (20 a 120 KHz) e um apenas para as baixas frequência, (20 a 120 Hz). O sistema europeu de TV digital usa o padrão MP2 (MPEG-1 Layer 2) [ISSO/ISC IS 11172-3 1993], mas algumas implementações seguem o padrão MPEG-2 AAC [ISSO/IEC 13818-7 1997] também adotado pelo sistema japonês.

O sistema brasileiro de TV digital adotou o padrão de sistema MPEG-4 para a codificação do áudio principal de um programa [ABNT NBR 15602-2 2007], com as características apresentadas na Tabela 1.

Tabela 1. Codificação de áudio no sistema brasileiro de TV digital.

	Receptores Fixos e Móveis	**Receptores Portáteis**
Padrão	ISSO/IEC 14496-3 (MPEG-4 AAC)	ISSO/IEC14496-3 (MPEG-4 AAC)
Nível@Perfil	ACC@L4 (para multicanal 5.1) HE-AAC v1@L4 (para estéreo)	HE-AAC v2@L3 (dois canais)
Taxa de amostragem	48 kHz	48 kHz

2.3 Codificação de Vídeo

Sabendo-se que um vídeo digital carrega muita informação redundante espacialmente (redundância intra-quadro) e temporalmente (redundância inter-quadros). Em um quadro de vídeo não ocorre, em geral, mudança abrupta de um pixel para um pixel consecutivo, exceto nos contornos dos objetos. Em particular, podemos empregar a codificação JPEG [ISSO/IEC 10918-1 1994] em cada quadro do vídeo para tirarmos proveito dessa redundância espacial. Essa técnica, aplica quadro a quadro, se constitui a base da codificação chamada MJPEG (Motion JPEG). Entretanto, ao empregarmos essa codificação, estaremos levando em conta apenas a redundância intra-quadro, quando a maior redundância pode estar nas informações contidas em quadros consecutivos (redundância inter-quadros), o que é explorado no padrão MPEG Vídeo.

Analisando a situação do sistema brasileiro se emprega uma técnica de codificação mais recente: o H.264 [ISSO/IEC 14496-10 2005], também conhecido como MPEG-4 Part. 10 ou MPEG-4 AVC (Advanced Video Coding). O objetivo do projeto h.264/AVC foi criar um padrão capaz de prover um vídeo de boa qualidade a uma taxa substancialmente mais baixa do que os padrões anteriores (MPEG-2 entre eles), sem aumentar muito sua complexidade, para facilitar uma implementação barata e eficiente. Um objetivo adicional era fornece flexibilidade suficiente para permitir sua aplicação em uma variedade de sistemas de taxas de transmissão alta e baixa, e também de resoluções de vídeo alta e baixa. H.264 contém várias facilidade novas que permitem uma compressão de vídeo muito mais eficiente e flexível. As técnicas empregadas fazem do H.264 um padrão significativamente melhor do que os padrões anteriores sob uma variedade de circunstancias e ambientes de aplicação em particular o ambiente de TV digital, onde oferece um desempenho radicalmente melhor do que o MPEG-2 vídeo, em especial nas situações de alta resolução e alta taxa de bits, quando, com a mesma qualidade de vídeo, gera uma taxa 50% ou menos do que a gerada pelo MPEG-2. Da mesma forma que o padrão MPEG-2 vídeo, H.264 é dividido em perfis e níveis. No caso do sistema brasileiro de TV digital são usados os perfis alto (HiP) para os receptores fixos e móveis e o perfil base (BP) para receptores portáteis [ABNT NBR 15602-1 2007], conforme indica a Tabela 2.

Tabela 2. Codificação de vídeo no sistema brasileiro de TV digital.

	Receptores Fixos e Móveis	**Receptores Portáteis**
Padrão	ITU-T H.264 (MPEG-4 AVC)	ITU-T H.264 (MPEG-4 AVC)
Nível@Perfil	HP@L4.0	BP@L1.3
Número de linhas do nível	480 (4:3 e 16:9), 720 (16:9), 1080(16:9)	SQVGA (160x120 ou 160x90), QVGA (320x240 ou 320x180) e CIF (352x288); todos em 4:3 e 16:9
Taxas de quadros	30 e 60 Hz	15 e 30 Hz

2.4 Middleware

Um programa não linear é enviado por difusão através de um carrossel de dados. O aplicativo, entre outras funções deve ser o responsável pela sincronização espacial e temporal dos vários objetos de mídia enviados no fluxo de transporte através do serviço assíncrono. Um aplicativo pode ser executado diretamente sobre o sistema operacional de um receptor. No entanto, os sistemas operacionais de propósito gerais não estão preparados para dar um bom suporte a aplicações de TV digital. Além disso, um aplicativo de TV desse ser capaz de ser executado em qualquer plataforma de hardware e sistema operacional.

Para tornar os aplicativos independentes da plataforma de hardware e software de um fabricante de receptor especifico, e para dar melhor suporte as aplicações voltadas para a TV, uma nova camada é acrescentada nos padrões de referência de um sistema de TV digital, e essa camada denominada middleware e seus padrões de referência do sistema brasileiro de TV digital terrestre tem seu próprio middleware de nome Ginga.

2.5 Requisitos

O middleware por sua vez tem a função de fornecer suporte ás aplicações. O suporte é fornecido através de uma interface de programação de aplicativos – API (Application Programming Interface), cuja funcionalidade oferecida deve ser regida pelas necessidades das aplicações a serem executadas no ambiente de TV digital. Dentre essas aplicações, obviamente, estão os programas não-lineares, foco principal de um sistema de TV digital. Levantar os requisitos das aplicações é, assim, levantar os requisitos de um middleware.

Em uma aplicação para TV, a interatividade deve ser usada com parcimônia. Em uma assistência coletiva, por exemplo, além da dificuldade imposta pelo ambiente mais pobre de interação, em comparação com aquela encontrada em um computador, o aparecimento de informações adicionais pela demanda de um telespectador pode aborrecer seu companheiro ao lado. O uso de dispositivos de exibição pessoais para interação (controle remoto com tela de baixa resolução,

celulares, PDAs etc.) poderia amenizar o problema. Isso nos leva a um segundo requisito a ser oferecido por um middleware que seria o suporte à múltiplos dispositivos de exibição.

Os receptores pessoais poderiam também comunicar-se entre si. Poderiam, por exemplo, permitir a inclusão de comentários textuais ou outros objetivos de mídia no aplicativo de TV recebido, e que o novo programa assim criado fosse distribuído aos demais participantes de uma comunidade (grupo de telespectadores), via canal de retorno, constituindo o que chamamos de TV social ou TV em comunidade.

A sincronização de objetos de mídia em um programa não linear em tempo de exibição é importante não só por possibilita a aplicação de TV em comunidade, onde a edição ao vivo é realizada no cliente telespectador, mas também por possibilitar a geração de programas não-lineares ao vivo pela emissora de TV. Alguns programas, a decisão de quais objetos da mídia comporão o programa pode ser decidida ao vivo. Isso nos leva a um terceiro requisito que deve ser oferecido por middleware que seria um suporte à edição ao vivo (em tempo de exibição).

Analisando a importância de um middleware tem se a adaptação do conteúdo e da forma como o conteúdo é exibido. Um conteúdo a ser exibido, por exemplo uma propaganda, deve poder ser adaptado ao tipo de usuário telespectador (por exemplo, se é um adulto ou uma criança), à localização do telespectador (por exemplo, uma propaganda diferenciada para cada loja mais próxima onde cada usuário telespectador poderia encontrar um certo produto), ao dispositivo exibidor (tamanho da tela, entre outras características) etc.

2.6 Ambiente de programação

As aplicações de TV digital são usualmente desenvolvidas usando dois paradigmas de programação distintos: o declarativo e o não-declarativo.

As linguagens de programação declarativas (linguagens que seguem ligadas a um declarativo) são linguagens de mais alto nível de abstração, usualmente ligadas a um domínio ou objetivo especifico. Nas linguagens declarativas, o

programador fornece apenas o conjunto das tarefas a serem realizadas, não estando preocupado com os detalhes de como o executor da linguagem (interpretador, compilador ou a própria máquina real ou virtual de execução) realmente programará essas tarefas. Linguagens declarativas resultam em uma declaração do resultado desejado, ao invés da sua decomposição em uma implementação algorítmica, e, portanto normalmente não necessitam de tantas linhas de código para definir certa tarefa. Entre as linguagens declarativas mais comuns estão a NCL (Nested Context Language) [ABNT NBR 15606-2 2007] e XHTML [W3C REC-xhtml1-20020801 2002].

Em uma programação não-declarativa, devemos informar cada passo a ser executado. Podendo afirmar que, em uma especificação segundo o paradigma não-declarativo, o programador possui um maior poder sobre o código sendo capaz de estabelecer todo o fluxo de controle e execução de seu programa. Entretanto, para isso, ele deve ser bem qualificado e conhecer bem os recursos de implementação. Linguagens não declarativas podem seguir diferentes modelos. Temos assim, as linguagens baseadas em módulos, orientadas a objetos etc. Entre as linguagens não-declarativas mais comuns no domínio da TV digital estão C, Java ECMAScript e Lua.

Todo o mundo de aplicações de um sistema de TV digital pode ser particionado em um conjunto de aplicações declarativas e um conjunto de aplicações não-declarativas. Uma aplicação declarativa é aquela onde o tipo de conteúdo da entidade inicial é declarativo, ou seja, é uma aplicação comandada pelo ambiente declarativo. Por outro lado, uma aplicação não-declarativa é aquela cujo tipo de conteúdo da entidade inicial é não-declarativo, ou seja, é uma aplicação comandada pelo ambiente procedural.

Praticamente todos os middlewares para TV digital terrestre oferecem suporte para desenvolvimento de aplicações usando os dois paradigmas de programação. Alguns middlewares só oferecem a suporte para aplicações declarativas (puras ou híbridas). Dá-se, informalmente, o nome de ambiente declarativo a esse suporte. Outros middlewares oferecem o suporte apenas a aplicação não-declarativas. Dá-

se, informalmente, o nome de ambiente procedural a esse suporte. A maioria dos middlewares, no entanto, são compostos por ambientes procedurais e declarativos.

Tabela 3. Ambiente de aplicações para receptores fixos e móveis.

Middleware	Sistema de TV	Ambiente Declarativo	Ambiente Procedural
Ginga	Brasileiro/SBTVD	Ginga-NCL [ABNT NBR 15606-2 2007] (linguagem declarativa = NCL, linguagem não declarativa = Lua)	Ginga-J (Linguagem não declarativa = Java)

Cabe enfatizar que tanto o ambiente declarativo quanto o procedural de um middleware deve dar suporte a todos os requisitos e cabe salientas que todos os middleware que apresentam os dois ambientes permitem que cada um deles use a facilidades do outro por meio de uma ponte definida através de APIs bilaterais padrão.

3 Linguagem NCL

As linguagens declarativas são baseadas em um modelo conceitual de dados. Um modelo conceitual deve representar os conceitos estruturais dos dados como os eventos e relacionamento entre os mesmos. O modelo deve também definir as regras estruturais e as operações sobre os dados para manipulação e atualização das estruturas

NCL (Linguagem de Contextos Aninhados) é uma linguagem de programação declarativa para desenvolvimento de aplicações para TV Digital. E por ser uma linguagem declarativa, não é necessário o conhecimento em lógica de programação. No entanto, quem já desenvolveu um site em HTML não terá dificuldades em entender a linguagem NCL, pois a estrutura é praticamente a mesma. E quem já tem algum conhecimento em XML, terá mais facilidade ainda para aprender a linguagem NCL. Até mesmo porque NCL é uma aplicação XML.

Baseada no modelo conceitual NCM, a NCL traz uma separação clara entre os conteúdos de mídia e a estrutura de uma aplicação. Um documento NCL apenas define como os objetos de mídia são estruturados e relacionados, no tempo e no espaço. Como uma linguagem de cola, ela não restringe nem prescreve os tipos de conteúdo dos objetos de mídia de uma aplicação.

Para definir todos esses objetos de mídia, a NCL faz uso de seu elemento <media>, como ilustrado na Listagem 1.

Listagem 1. Elementos <media>.

```
 <media id="animation" src="../media/animGar.mp4"/>
<media id="choro" src="../media/choro.mp3"/>
<media id="drible" src="../media/drible.mp4"/>
<media id="photo" src="../media/photo.png"/>
```

Note que todo elemento NCL deve começar com o caractere “<”. Quando o elemento não tem nenhum conteúdo nem elementos filhos, ele deve terminar sempre com “/>”, como na Listagem 1. Em caso contrário, a definição dos atributos do elemento termina com o caractere “>”. Após a definição de seu conteúdo ou

elementos filhos, o elemento termina repetindo seu nome envolvido com os caracteres “</” na frente e o caractere “>” atrás, como visto a seguir, na redefinição do elemento <media id=“animation”.../> da Listagem 1 que, na Listagem 2 representa o vídeo da animação com dois elementos filhos aninhados, denominados <area>.

Listagem 2. Elementos <area>.

```
<media id="animation" src="../media/anima.mp4" descriptor="screenDesc">
        <area id="segDrible" begin="12s"/>
        <area id="segPhoto" begin="41s"/>
</media>
```

Nosso próximo passo no projeto da aplicação é definir em que posição da tela os vários objetos de mídia serão exibidos. A visão de leiaute desejada. São definidas regiões para exibição do vídeo da animação (“screenReg”), em tela cheia, e uma região para exibição do vídeo do drible e da foto (“frameReg”).

A definição dos espaços de exibição pode ser realizada por meio de elementos <property>, filhos dos elementos <media> que representam cada objeto de mídia da aplicação, como ilustrado na Listagem 3.

Listagem 3. Elementos <property>.

```
<media id="animation" src="../media/anima.mp4">
        <area id="segDrible" begin="11.5s"/>
        <area id="segPhoto" begin="41s"/>
        <property name="width" value="100%"/>
        <property name="heigth" value="100%"/>
        <property name="zIndex" value="2"/>
</media>
<media id="choro" src="../media/choro.mp3"/>
<media id="drible" src="../media/drible.mp4">
        <property name="left" value="5%"/>
        <property name="top" value="6.7%"/>
        <property name="width" value="18.5%"/>
        <property name="heigth" value="18.5%"/>
        <property name="zIndex" value="3"/>
</media>
<media id="photo" src="../media/photo.png">
        <property name="left" value="5%"/>
        <property name="top" value="6.7%"/>
        <property name="width" value="18.5%"/>
```

```
        <property name="heigth" value="18.5%"/>
        <property name="zIndex" value="3"/>
        <property name="explicitDur" value="5s"/>
</media>
```

O elemento <body> é único em um documento NCL e pode ter um ou mais elementos <port> como filho. O elemento <port> indica por onde pode começar uma exibição do documento. No nosso exemplo, como ilustrado na Listagem 4, a exibição começa sempre pela apresentação do vídeo da animação.

Listagem 4. Elementos <body> e <port>.

```
<body>
        <port id="entry" component="animation"/>
        <!—definição dos diversos elementos de <media> do documento -->
        <!—definição dos diversos relacionamentos, elementos
                                                        <link> do documento -->
</body>
```

Relacionamentos são definidos por elementos <link>, como ilustrado na Listagem 5, para nosso exemplo do aplicativo. O relacionamento ilustrado define que, ao ser iniciada a exibição do vídeo da animação, o áudio com o chorinho também deve ser iniciado, mas 5 segundos depois (como determinamos no enunciado do exemplo, esse objeto de mídia é apresentado após os créditos iniciais do vídeo da animação, que duram 5 segundos).

Listagem 5. Elementos <link> e <bind>.

```
 <link id="lMusic" xconnector="onBeginStart_delay">
        <bind role="onBegin" component="animation"/>
        <bind role="start" component="choro">
              <bindParam name="delay" value="5s"/>
        </bind>
</link>
```

No relacionamento definido pelo elemento <link> “lMusic” da Listagem 3.5, a relação referenciada é definida pelo elemento <causalConnector>, filho do elemento <connectorBase>, conforme ilustra a Listagem 6. Na relação (Listagem 5), o papel “onBegin” do conector é associado ao componente “animation”, que, como vimos na Listagem 2, é associado ao vídeo da animação, definindo que, ao começar a

exibição do vídeo, deve-se dar partida (*role* = “start”) à exibição do chorinho, mas 5 segundos a partir do início do vídeo.

Listagem 6. Elemento <causalConnector> e seus elementos filhos.

```
<connectorBase>
        <causalConnector id="onBeginStart_delay">
        <connectorParam name="delay"/>
        <simpleCondition role="onBegin"/>
        <simpleAction role="start" delay="$delay" max="unbounded" qualifier="par"/>
        </causalConnector>
</connectorBase>
```

Bases de conectores são definidas como elemento filho do elemento <head>, que define, como veremos, as partes reusáveis de uma aplicação. No caso, um mesmo conector pode ser usado por mais de um elemento <link>. A Listagem 7 apresenta a definição completa do elemento <head> da aplicação exemplo <head> .

Listagem 7. Elemento <head> e seus elementos filhos.

```
<connectorBase>
        <causalConnector id="onBeginStart_delay">
           <connectorParam name="delay"/>
           <simpleCondition role="onBegin"/>
          <simpleAction role="start" delay="$delay" max="unbounded" qualifier="par"/>
        </causalConnector>
        <causalConnector id="onBeginStart">
           <simpleCondition role="onBegin"/>
          <simpleAction role="start" max="unbounded" qualifier="par"/>
        </causalConnector>
        <causalConnector id="onEndStop">
                <simpleCondition role="onEnd"/>
               <simpleAction role="stop" max="unbounded" qualifier="par"/>
        </causalConnector>
</connectorBase>
```

Podemos agora definir todo o elemento <body> e seus filhos, para o exemplo, apresentados na Listagem 8.

Listagem 8. Elemento <body> e seus elementos filhos.

```
<body>
        <port id="entry" component="animation"/>
        <media id="animation" src="../media/animGar.mp4">
          <area id="segDrible" begin="11.5s"/>
```

```
        <area id="segPhoto" begin="41s"/>
        <property name="width" value="100%"/>
        <property name="heigth" value="100%"/>
        <property name="zIndex" value="2"/>
      </media>
      <media id="choro" src="../media/choro.mp3"/>
      <media id="drible" src="../media/drible.mp4">
         <property name="left" value="5%"/>
         <property name="top" value="6.7%"/>
        <property name="width" value="18.5%"/>
        <property name="heigth" value="18.5%"/>
        <property name="zIndex" value="3"/>
      </media>
      <media id="photo" src="../media/photo.png">
        <property name="left" value="5%"/>
       <property name="top" value="6.7%"/>
       <property name="width" value="18.5%"/>
       <property name="heigth" value="18.5%"/>
      <property name="zIndex" value="3"/>
      <property name="explicitDur" value="5s"/>
      </media>
      <link id="lMusic" xconnector="onBeginStart_delay">
        <bind role="onBegin" component="animation"/>
        <bind role="start" component="choro">
           <bindParam name="delay" value="5s"/>
       </bind>
     </link>
      <link id="lDrible" xconnector="onBeginStart">
        <bind role="onBegin" component="animation" interface="segDrible"/>
        <bind role="start" component="drible"/>
      </link>
      <link id="lPhoto" xconnector="onBeginStart">
         <bind role="onBegin" component="animation" interface="segPhoto"/>
        <bind role="start" component="photo"/>
      </link>
      <link id="lEnd" xconnector="onEndStop">
         <bind role="onEnd" component="animation"/>
         <bind role="stop" component="choro"/>
      </link>
</body>
```

A Listagem 9 ilustra a aplicação completa. Note que toda aplicação NCL começa com a linha: <?xml version="1.0" encoding="ISO-8859-1"?> seguida da especificação do elemento <ncl>, onde o atributo *id* define um identificador para a aplicação e o atributo *xmlns* define o espaço de nomes (namespace) onde estão

definidos os esquemas do perfil NCL EDTV, para verificação, pelo parser XML, da validade da aplicação.

Listagem 9. Documento NCL.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Exemplo de sincronismo sem a interacao do usuario e com reúso apenas
de relações-->
<ncl id="syncEx" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
<head>
<connectorBase>
   <causalConnector id="onBeginStart_delay">
      <connectorParam name="delay"/>
      <simpleCondition role="onBegin"/>
      <simpleAction role="start" delay="$delay" max="unbounded"
                                                    qualifier="par"/>
   </causalConnector>
   <causalConnector id="onBeginStart">
       <simpleCondition role="onBegin"/>
       <simpleAction role="start" max="unbounded" qualifier="par"/>
   </causalConnector>
   <causalConnector id="onEndStop">
       <simpleCondition role="onEnd"/>
       <simpleAction role="stop" max="unbounded" qualifier="par"/>
  </causalConnector>
</connectorBase>
</head>
<body>
   <port id="entry" component="animation"/>
   <media id="animation" src="../media/animGar.mp4">
     <area id="segDrible" begin="11.5s"/>
     <area id="segPhoto" begin="41s"/>
     <property name="width" value="100%"/>
    <property name="heigth" value="100%"/>
    <property name="zIndex" value="2"/>
   </media>
   <media id="choro" src="../media/choro.mp3"/>
   <media id="drible" src="../media/drible.mp4">
      <property name="left" value="5%"/>
      <property name="top" value="6.7%"/>
      <property name="width" value="18.5%"/>
      <property name="heigth" value="18.5%"/>
     <property name="zIndex" value="3"/>
   </media>
   <media id="photo" src="../media/photo.png">
      <property name="left" value="5%"/>
      <property name="top" value="6.7%"/>
      <property name="width" value="18.5%"/>
```

```
      <property name="heigth" value="18.5%"/>
      <property name="zIndex" value="3"/>
      <property name="explicitDur" value="5s"/>
  </media>
<link id="lMusic" xconnector="onBeginStart_delay">
   <bind role="onBegin" component="animation"/>
      <bind role="start" component="choro">
      <bindParam name="delay" value="5s"/>
    </bind>
</link>
<link id="lDrible" xconnector="onBeginStart">
    <bind role="onBegin" component="animation" interface="segDrible"/>
    <bind role="start" component="drible"/>
</link>
<link id="lPhoto" xconnector="onBeginStart">
    <bind role="onBegin" component="animation" interface="segPhoto"/>
    <bind role="start" component="photo"/>
</link>
<link id="lEnd" xconnector="onEndStop">
    <bind role="onEnd" component="animation"/>
    <bind role="stop" component="choro"/>
</link>
</body>
</ncl>
```

3.1 Aplicações desenvolvidas contidas na Mídia Externa

O Clube NCL é um repositório de aplicações interativas onde autores podem publicar suas idéias, talentos e suas técnicas de desenvolvimento usando a linguagem NCL e scripts Lua. Todo o site é basicamente um ambiente colaborativo onde todos podem participar. Abaixo segue a descrição:

- **Tv digital quis:** Quis é uma forma de avaliar uma grande quantidade de pessoas com um questionário com respostas do tipo "certo ou errado" e chegar em um consenso geral.
- **Slide Show:** Visualiza fotos e imagens na TV DIGITAL.
- **Jogo Da velha:** Jogo da Velha através da TV Digital.
- **Porta Retrato:** Visualiza fotos com molduras enquanto estiver em stand-by

- **RSS:** Anuncio de notícias em tempo real do site enquanto estiver interagindo com a TV digital
- **Twit:** Visualiza mensagens do twiter na sua TV digital.

4 Aplicações desenvolvidas

Usando os conhecimentos adquiridos acima obtivemos as seguinte aplicação

4.1 Tv digital quis:

Imagem quis TV digital 4.1.1

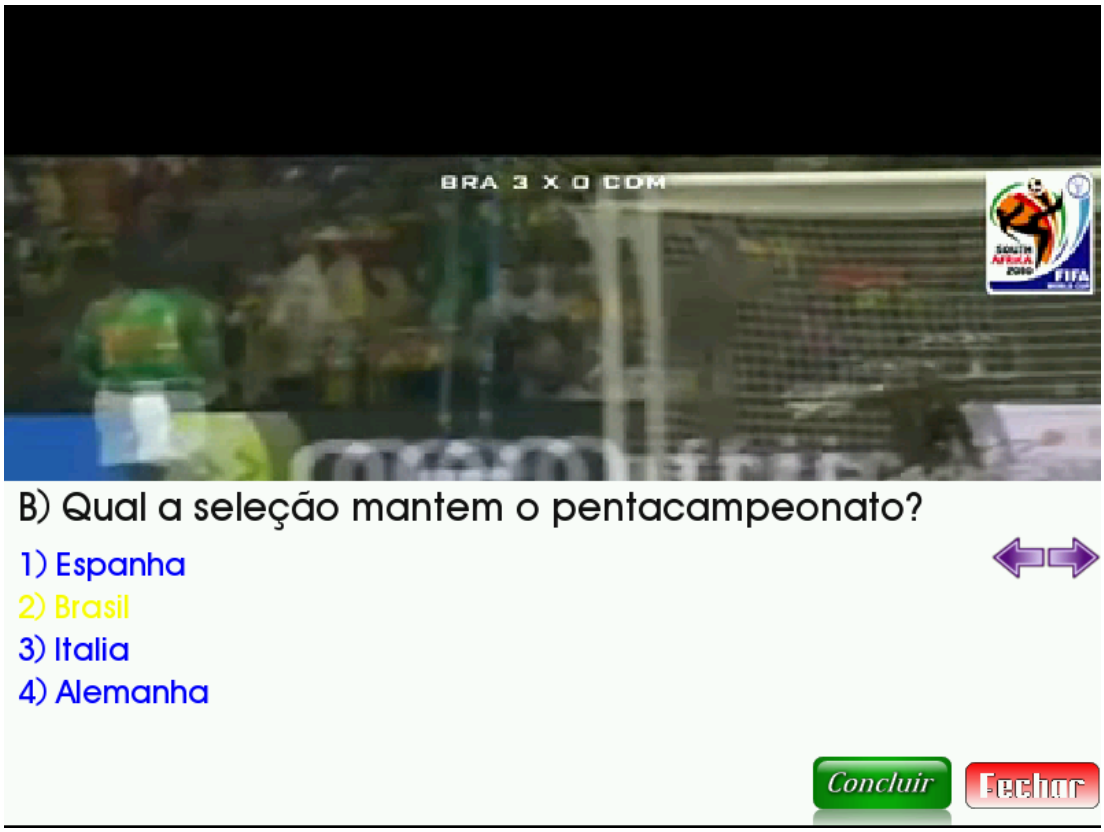


Imagem quis TV digital 4.1.2

Código fonte quis, main.ncl 4.1.3

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="main" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
	<head>
		<transitionBase>
			<transition id="tFade" type="fade"/>
		</transitionBase>

		<regionBase>
	 <region id="rgVideo" width="100%" height="100%" zIndex="0">
			 <region id="rgLua" width="100%" height="42%" left="0"
top="58%" zIndex="1" />
			</region>
		</regionBase>

		<descriptorBase>
			<descriptor id="dLua" region="rgLua" focusIndex="luaIdx"
transIn="tFade" transOut="tFade" />
```

```
                    <descriptor id="dVideo" region="rgVideo" />
                </descriptorBase>

                <connectorBase>
            <causalConnector id="onBeginStart">
               <simpleCondition role="onBegin"/>
               <simpleAction role="start"/>
            </causalConnector>

            <causalConnector id="onKeySelectionSet">
                    <connectorParam name="key"/>
                    <connectorParam name="value"/>

               <simpleCondition role="onSelection" key="$key"/>
               <simpleAction role="set" value="$value"/>
            </causalConnector>

                    <causalConnector id="onEndStart">
                        <simpleCondition role="onEnd"/>
                        <simpleAction role="start"/>
                    </causalConnector>

                    <causalConnector id="onKeySelectionStop">
                        <connectorParam name="key"/>
                        <simpleCondition role="onSelection" key="$key"/>
                        <simpleAction role="stop"/>
                    </causalConnector>

                    <causalConnector id="onBeginStop">
                        <simpleCondition role="onBegin"/>
                        <simpleAction role="stop"/>
                    </causalConnector>

                </connectorBase>
            </head>

            <body>
                <port id="pVideo" component="video1"/>

           <media type="application/x-ginga-settings" id="programSettings">
              <property name="service.currentKeyMaster" value="luaIdx"/>
           </media>


           <!--Vídeo sob licença Creative Commons, obtido em
    http://creativecommons.org/video -->
                <media id="video1" src="media/Wanna_Work_Together_-
    _Creative_Commons.avi" descriptor="dVideo">
```

```
        </media>

        <media id="lua" src="main.lua" descriptor="dLua">
            <property name="finalizar"/>
        </media>

    <link xconnector="onBeginStart">
            <bind role="onBegin" component="video1" />
            <bind role="start" component="lua" />
    </link>

    <link xconnector="onKeySelectionSet">
        <bind role="onSelection" component="video1">
            <bindParam name="key" value="GREEN"/>
          </bind>

        <bind role="set" component="lua" interface="finalizar">
            <bindParam name="value" value="true"/>
          </bind>
    </link>

        <link xconnector="onEndStart">
            <bind component="video1" role="onEnd" />
            <bind component="video1" role="start" />

        </link>

        <link xconnector="onKeySelectionStop">
            <bind component="video1" role="onSelection">
                <bindParam name="key" value="RED"/>

            </bind>
            <bind component="lua" role="stop" />
        </link>
        </body>
</ncl>
```

Código fonte quis, perguntas.lua 4.1.4

```
perguntas = {
    {
      per = "Qual o nome do atual técnico da seleção brasileira?",
      resp = {"Filipão", "Mano Menezes", "Dunga", "Tite"},
      corr = 1
    },
```

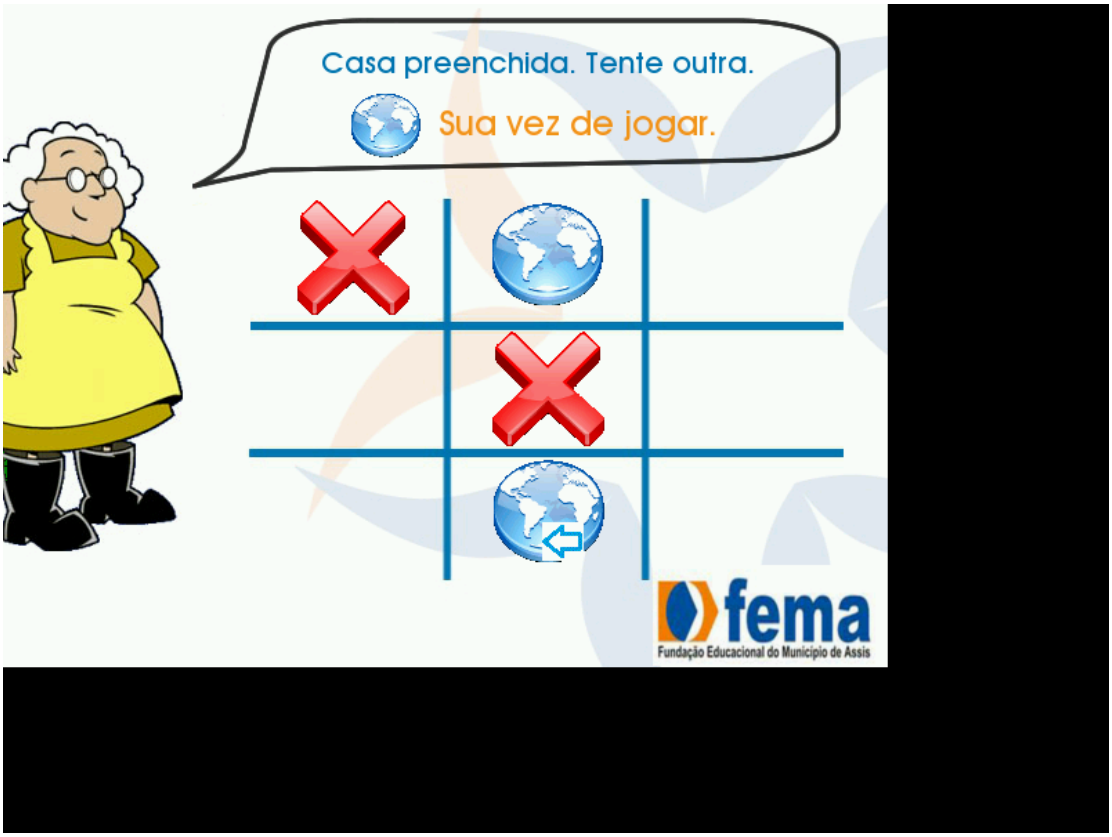
```
    {
      per = "Qual a seleção mantem o pentacampeonato?",
      resp = {"Espanha", "Brasil", "Italia", "Alemanha"},
      corr = 2
    },

    {
      per = "Quem ficou de fora da Copa das Confederações 2013?",
      resp = { "Julio Cesar", "Huck", "Neimar", "Ronaldinho Gaúcho"},
      corr = 4
    },
}
```

4.2 Jogo da Velha

Imagem Jogo da Velha TV digital 4.2.1

Imagem Jogo da Velha TV digital 4.2.2



Código fonte Jogo da Velha, main.ncl 4.2.3

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="nclClicks" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">

<head>

<regionBase>
        <region id="rgLua" left="0" top="0" width="640" height="480" zIndex="2" />
        <region id="rgfundo" left="0" top="0" width="640" height="480" zIndex="1" />
<!--    <region id="rbotaoIniciar" left="247" top="368" width="379" height="40"
zIndex="2" /> -->
        <region id="rbotaoSair" left="24" top="442" width="316" height="29"
zIndex="3" />
        <region id="rbotaoJogarSozinho" left="24" top="364" width="371" height="35"
zIndex="4" />
        <region id="rbotaoJogarAmigo" left="24" top="403" width="432" height="35"
zIndex="5" />
        <region id="rnivelDificuldade" left="14" top="350" width="432" height="27"
zIndex="6" />
        <region id="rbotaoFacil" left="137" top="379" width="164" height="50"
zIndex="7" />
```

```
        <region id="rbotaoDificil" left="137" top="426" width="183" height="50"
zIndex="8" />
</regionBase>

<descriptorBase>
        <descriptor id="dsLua" region="rgLua" focusIndex="2" />
        <descriptor id="dfundo" region="rgfundo" focusIndex="1" />
        <!-- <descriptor id="dbotaoIniciar" region="rbotaoIniciar" focusIndex="3"  /> --
>
        <descriptor id="dbotaoSair" region="rbotaoSair"  />
        <descriptor id="dbotaoJogarSozinho" region="rbotaoJogarSozinho"  />
        <descriptor id="dbotaoJogarAmigo" region="rbotaoJogarAmigo"  />
        <descriptor id="dnivelDificuldade" region="rnivelDificuldade"  />
        <descriptor id="dbotaoFacil" region="rbotaoFacil"  />
        <descriptor id="dbotaoDificil" region="rbotaoDificil"  />
</descriptorBase>

<connectorBase>

        <causalConnector id="onBeginStartN">
                <simpleCondition role="onBegin"/>
                <simpleAction role="start" max="unbounded" qualifier="seq" />
        </causalConnector>

        <causalConnector id="onBeginStopN">
                <simpleCondition role="onBegin"/>
                <simpleAction role="stop" max="unbounded" qualifier="seq" />
        </causalConnector>

        <causalConnector id="onEndStartN">
                <simpleCondition role="onEnd"/>
                <simpleAction role="start" max="unbounded" qualifier="par" />
        </causalConnector>

        <causalConnector id="onEndStopN">
                <simpleCondition role="onEnd"/>
                <simpleAction role="stop" max="unbounded" qualifier="par" />
        </causalConnector>

        <causalConnector id="onKeySelectionStopNStartNSetN">
                <connectorParam name="keyCode" />
                <connectorParam name="var"/>
                <simpleCondition role="onSelection" key="$keyCode"/>
                <compoundAction operator="seq">
                        <simpleAction role="stop" max="unbounded" qualifier="seq"/>
                        <simpleAction role="start" max="unbounded" qualifier="par"/>
                        <simpleAction role="set" value="$var" max="unbounded"
qualifier="par" />
```

```
            </compoundAction>
        </causalConnector>

        <causalConnector id="onKeySelectionStopNStartN">
            <connectorParam name="keyCode" />
            <connectorParam name="var"/>
            <simpleCondition role="onSelection" key="$keyCode"/>
            <compoundAction operator="seq">
                <simpleAction role="stop" max="unbounded" qualifier="seq"/>
                <simpleAction role="start" max="unbounded" qualifier="par"/>
            </compoundAction>
        </causalConnector>

        <causalConnector id="onKeySelectionStopN">
            <connectorParam name="keyCode" />
            <connectorParam name="var"/>
            <simpleCondition role="onSelection" key="$keyCode"/>
            <simpleAction role="stop" max="unbounded" qualifier="seq"/>
        </causalConnector>

        <causalConnector id="onBeginStartNSetN">
            <simpleCondition role="onBegin"/>
            <connectorParam name="var"/>
            <compoundAction operator="seq">
                <simpleAction role="start" max="unbounded" qualifier="par"/>
                <simpleAction role="set" value="$var" max="unbounded"
qualifier="par" />
            </compoundAction>
        </causalConnector>

</connectorBase>

</head>
<body>
        <port id="init" component="fundo"/>

<!--
        Nos de propriedade
-->
        <media id="programSettings" type="application/x-ginga-settings" >
            <property name="service.currentKeyMaster" value="1" />
        </media>

<!--
        <media id="programSettings2" type="application/x-ginga-settings" >
            <property name="service.currentFocus" value="1" />
        </media>
-->
```

```
<!--
	Nos de midia
-->
	<media id="lua" type="application/x-ginga-NCLua" src="main.lua" descriptor="dsLua" >
		<area id="ReStart" label="ReStart"/>
		<property name="nivel" />
	</media>

	<media id="fundo" type="image/jpeg" src="media/fundo.jpg" descriptor="dfundo" />
	<media id="somJogo" type="audio/mp3" src="media/somJogo.mp3"  />
	<!-- <media id="botaoIniciar" type="image/jpeg" src="media/botao_iniciar.jpg" descriptor="dbotaoIniciar" /> -->
	<media id="botaoJogarSozinho" type="image/jpeg" src="media/botao_jogar_sozinho.jpg" descriptor="dbotaoJogarSozinho" />
	<media id="botaoJogarAmigo" type="image/jpeg" src="media/botao_jogar_amigo.jpg" descriptor="dbotaoJogarAmigo" />
	<media id="botaoSair" type="image/jpeg" src="media/botao_sair.jpg" descriptor="dbotaoSair" />
	<media id="nivelDificuldade" type="image/jpeg" src="media/nivel_dificuldade.jpg" descriptor="dnivelDificuldade" />
	<media id="botaoFacil" type="image/jpeg" src="media/botao_facil.jpg" descriptor="dbotaoFacil" />
	<media id="botaoDificil" type="image/jpeg" src="media/botao_dificil.jpg" descriptor="dbotaoDificil" />


<!--
	Tela de abertura
-->
	<link xconnector="onBeginStartN">
		<bind role="onBegin" component="fundo"/>
		<bind role="start" component="botaoJogarSozinho"/>
		<bind role="start" component="botaoJogarAmigo"/>
		<bind role="start" component="botaoSair"/>
	</link>

	<link xconnector="onEndStopN">
		<bind role="onEnd" component="fundo"/>
		<bind role="stop" component="botaoJogarSozinho"/>
		<bind role="stop" component="botaoJogarAmigo"/>
		<bind role="stop" component="botaoSair"/>
	</link>


<!--
```

```
        Inicio do jogo com amigo
-->
        <link xconnector="onKeySelectionStopNStartNSetN">
                <linkParam name="keyCode" value="GREEN" />
                <bind component="botaoJogarAmigo" role="onSelection" />
                <bind role="stop" component="fundo"/>
                <bind role="stop" component="botaoJogarSozinho"/>
                <bind role="stop" component="botaoJogarAmigo"/>
                <bind role="stop" component="botaoSair"/>
                <bind role="start" component="lua"/>
        <bind                component="programSettings"                role="set"
interface="service.currentKeyMaster">
                                <bindParam name="var" value="2"/>
                </bind>
        </link>

        <link xconnector="onBeginStartN">
                <bind role="onBegin" component="lua"/>
                <bind role="start" component="somJogo"/>
        </link>

        <link xconnector="onEndStopN">
                <bind role="onEnd" component="lua"/>
                <bind role="stop" component="somJogo"/>
        </link>

<!--
        <link xconnector="onBeginStopN">
                <bind role="onBegin" component="lua"/>
                <bind role="stop" component="botaoJogarSozinho"/>
                <bind role="stop" component="botaoJogarAmigo"/>
                <bind role="stop" component="botaoSair"/>
        </link>
-->

<!--
        Tela para escolha de dificuldade do jogo Sozinho
-->

        <link xconnector="onKeySelectionStopNStartN">
                <linkParam name="keyCode" value="BLUE" />
                <bind component="botaoJogarSozinho" role="onSelection" />
                <bind role="stop" component="botaoJogarAmigo"/>
                <bind role="stop" component="botaoJogarSozinho"/>
                <bind role="stop" component="botaoSair"/>
                <bind role="start" component="botaoFacil"/>
                <bind role="start" component="botaoDificil"/>
                <bind role="start" component="nivelDificuldade"/>
```

```
        </link>
<!--
        <link xconnector="onBeginStopN">
                <bind role="onBegin" component="nivelDificuldade"/>
                <bind role="stop" component="botaoJogarSozinho"/>
        </link>
        -->


<!--
        Inicio do jogo Sozinho
-->
        <link xconnector="onKeySelectionStopNStartNSetN">
                <linkParam name="keyCode" value="1" />
                <bind component="botaoFacil" role="onSelection" />
                <bind role="stop" component="fundo"/>
                <bind role="stop" component="nivelDificuldade"/>
                <bind role="stop" component="botaoDificil"/>
                <bind role="stop" component="botaoFacil"/>
                <bind role="start" component="lua"/>
                <bind component="programSettings" role="set"
interface="service.currentKeyMaster">
                        <bindParam name="var" value="2"/>
                </bind>
                <bind component="lua" role="set" interface="nivel">
                        <bindParam name="var" value="1"/>
                </bind>
        </link>

        <link xconnector="onKeySelectionStopNStartNSetN">
                <linkParam name="keyCode" value="2" />
                <bind component="botaoFacil" role="onSelection" />
                <bind role="stop" component="fundo"/>
                <bind role="stop" component="nivelDificuldade"/>
                <bind role="stop" component="botaoDificil"/>
                <bind role="stop" component="botaoFacil"/>
                <bind role="start" component="lua"/>
                <bind component="programSettings" role="set"
interface="service.currentKeyMaster">
                        <bindParam name="var" value="2"/>
                </bind>
                <bind component="lua" role="set" interface="nivel">
                        <bindParam name="var" value="2"/>
                </bind>
        </link>
```

```
<!--
        Elos responsaveis pelo reinicio do jogo
-->
        <link xconnector="onBeginStartNSetN">
                <bind role="onBegin" component="lua" interface="ReStart"/>
                <bind component="programSettings" role="set"
interface="service.currentKeyMaster">
                        <bindParam name="var" value="1"/>
                </bind>
        </link>

        <link xconnector="onEndStartN">
                <bind role="onEnd" component="lua" />
                <bind role="start" component="fundo"/>
        </link>

<!--
        Elos responsaveis por sair do jogo
-->
        <link xconnector="onKeySelectionStopN">
                <linkParam name="keyCode" value="RED" />
                <bind component="botaoSair" role="onSelection" />
                <bind role="stop" component="fundo"/>
                <bind role="stop" component="botaoJogarSozinho"/>
                <bind role="stop" component="botaoJogarAmigo"/>
                <bind role="stop" component="lua"/>
        </link>

        </body>
</ncl>
```

Código fonte Jogo da Velha, intelligence-class.lua 4.2.4

```
Intelligence = {}

function Intelligence:new()
        obj = {}
        -- heranca
        setmetatable(obj,self)
        self.__index = self

        return obj
end

function Intelligence:verificaVence(tabuleiro)
        local vence = {} --tabela onde e armazenado a jogada vencedora
```

```
        if tabuleiro:isCampoVazio(1,1) then
                print ("campo 1, 1 vazio - x")
                if tabuleiro:isPreenchido(1,2,XIS) and tabuleiro:isPreenchido(1,3,XIS)
then
                        vence.jogada = 1
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,1,XIS)                 and
tabuleiro:isPreenchido(3,1,XIS) then
                        vence.jogada = 1
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,2,XIS)                 and
tabuleiro:isPreenchido(3,3,XIS) then
                        vence.jogada = 1
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(1,2) then
                print ("campo 1, 2 vazio - x")
                if tabuleiro:isPreenchido(1,3,XIS) and tabuleiro:isPreenchido(1,1,XIS)
then
                        vence.jogada = 2
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,2,XIS)                 and
tabuleiro:isPreenchido(3,2,XIS) then
                        vence.jogada = 2
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(1,3) then
                print ("campo 1, 3 vazio - x")
                if tabuleiro:isPreenchido(2,3,XIS) and tabuleiro:isPreenchido(3,3,XIS)
then
                        vence.jogada = 3
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,2,XIS)                 and
tabuleiro:isPreenchido(3,1,XIS) then
                        vence.jogada = 3
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(1,1,XIS)                 and
tabuleiro:isPreenchido(1,2,XIS) then
                        vence.jogada = 3
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(2,1) then
```

```
                print ("campo 2, 1 vazio - x")
                if tabuleiro:isPreenchido(3,1,XIS) and tabuleiro:isPreenchido(1,1,XIS)
then
                        vence.jogada = 4
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,2,XIS)                 and
tabuleiro:isPreenchido(2,3,XIS) then
                        jogaMax = 4
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(2,2) then
                print ("campo 2, 2 vazio - x")
                if tabuleiro:isPreenchido(3,2,XIS) and tabuleiro:isPreenchido(1,2,XIS)
then
                        vence.jogada = 5
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,3,XIS)                 and
tabuleiro:isPreenchido(2,1,XIS) then
                        vence.jogada = 5
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(3,1,XIS)                 and
tabuleiro:isPreenchido(1,3,XIS) then
                        vence.jogada = 5
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(3,3,XIS)                 and
tabuleiro:isPreenchido(1,1,XIS) then
                        vence.jogada = 5
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(2,3) then
                print ("campo 2, 3 vazio - x")
                if tabuleiro:isPreenchido(3,3,XIS) and tabuleiro:isPreenchido(1,3,XIS)
then
                        vence.jogada = 6
                        vence.XIS = true
                elseif                  tabuleiro:isPreenchido(2,1,XIS)                 and
tabuleiro:isPreenchido(2,2,XIS) then
                        vence.jogada = 6
                        vence.XIS = true
                end
        end

        if tabuleiro:isCampoVazio(3,1) then
                print ("campo 3, 1 vazio - x")
```

```
            if tabuleiro:isPreenchido(3,2,XIS) and tabuleiro:isPreenchido(3,3,XIS)
then
                vence.jogada = 7
                vence.XIS = true
            elseif tabuleiro:isPreenchido(1,3,XIS) and
tabuleiro:isPreenchido(2,2,XIS) then
                vence.jogada = 7
                vence.XIS = true
            elseif tabuleiro:isPreenchido(1,1,XIS) and
tabuleiro:isPreenchido(2,1,XIS) then
                vence.jogada = 7
                vence.XIS = true
            end
        end

        if tabuleiro:isCampoVazio(3,2) then
            print ("campo 3, 2 vazio - x")
            if tabuleiro:isPreenchido(3,3,XIS) and tabuleiro:isPreenchido(3,1,XIS)
then
                vence.jogada = 8
                vence.XIS = true
            elseif tabuleiro:isPreenchido(1,2,XIS) and
tabuleiro:isPreenchido(2,2,XIS) then
                vence.jogada = 8
                vence.XIS = true
            end
        end

        if tabuleiro:isCampoVazio(3,3) then
            print ("campo 3, 3 vazio - x")
            if tabuleiro:isPreenchido(1,1,XIS) and tabuleiro:isPreenchido(2,2,XIS)
then
                vence.jogada = 9
                vence.XIS = true
            elseif tabuleiro:isPreenchido(3,1,XIS) and
tabuleiro:isPreenchido(3,2,XIS) then
                vence.jogada = 9
                vence.XIS = true
            elseif tabuleiro:isPreenchido(1,3,XIS) and
tabuleiro:isPreenchido(2,3,XIS) then
                vence.jogada = 9
                vence.XIS = true
            end
        end

        return vence
end
```

```
function Intelligence:verificaPerde(tabuleiro)
	local perde = {}

	if tabuleiro:isCampoVazio(1,1) then
		print ("campo 1, 1 vazio - b")
		if tabuleiro:isPreenchido(1,2,BOLA) and
tabuleiro:isPreenchido(1,3,BOLA) then
			perde.jogada = 1;
			perde.XIS = true;
		elseif tabuleiro:isPreenchido(2,1,BOLA) and
tabuleiro:isPreenchido(3,1,BOLA) then
			perde.jogada = 1;
			perde.XIS = true;
		elseif tabuleiro:isPreenchido(2,2,BOLA) and
tabuleiro:isPreenchido(3,3,BOLA) then
			perde.jogada = 1;
			perde.XIS = true;
		end
	end

	if tabuleiro:isCampoVazio(1,2) then
		print ("campo 1, 2 vazio - b")
		if tabuleiro:isPreenchido(1,3,BOLA) and
tabuleiro:isPreenchido(1,1,BOLA) then
			perde.jogada = 2;
			perde.XIS = true;
		elseif tabuleiro:isPreenchido(2,2,BOLA) and
tabuleiro:isPreenchido(3,2,BOLA) then
			perde.jogada = 2;
			perde.XIS = true;
		end
	end

	if tabuleiro:isCampoVazio(1,3) then
		print ("campo 1, 3 vazio - b")
		if tabuleiro:isPreenchido(2,3,BOLA) and
tabuleiro:isPreenchido(3,3,BOLA) then
			perde.jogada = 3;
			perde.XIS = true;
		elseif tabuleiro:isPreenchido(2,2,BOLA) and
tabuleiro:isPreenchido(3,1,BOLA) then
			perde.jogada = 3;
			perde.XIS = true;
		elseif tabuleiro:isPreenchido(1,1,BOLA) and
tabuleiro:isPreenchido(1,2,BOLA) then
			perde.jogada = 3;
			perde.XIS = true;
		end
```

```
        end

        if tabuleiro:isCampoVazio(2,1) then
                print ("campo 2, 1 vazio - b")
                if                tabuleiro:isPreenchido(3,1,BOLA)                and
tabuleiro:isPreenchido(1,1,BOLA) then
                        perde.jogada = 4;
                        perde.XIS = true;
                elseif                tabuleiro:isPreenchido(2,2,BOLA)                and
tabuleiro:isPreenchido(2,3,BOLA) then
                        jogaMax = 4;
                        perde.XIS = true;
                end
        end

        if tabuleiro:isCampoVazio(2,2) then
                print ("campo 2, 2 vazio - b")
                if                tabuleiro:isPreenchido(3,2,BOLA)                and
tabuleiro:isPreenchido(1,2,BOLA) then
                        perde.jogada = 5;
                        perde.XIS = true;
                elseif                tabuleiro:isPreenchido(2,3,BOLA)                and
tabuleiro:isPreenchido(2,1,BOLA) then
                        perde.jogada = 5;
                        perde.XIS = true;
                elseif                tabuleiro:isPreenchido(3,1,BOLA)                and
tabuleiro:isPreenchido(1,3,BOLA) then
                        perde.jogada = 5;
                        perde.XIS = true;
                elseif                tabuleiro:isPreenchido(3,3,BOLA)                and
tabuleiro:isPreenchido(1,1,BOLA) then
                        perde.jogada = 5;
                        perde.XIS = true;
                end
        end

        if tabuleiro:isCampoVazio(2,3) then
                print ("campo 2, 3 vazio - b")
                if                tabuleiro:isPreenchido(3,3,BOLA)                and
tabuleiro:isPreenchido(1,3,BOLA) then
                        perde.jogada = 6;
                        perde.XIS = true;
                elseif                tabuleiro:isPreenchido(2,1,BOLA)                and
tabuleiro:isPreenchido(2,2,BOLA) then
                        perde.jogada = 6;
                        perde.XIS = true;
                end
        end
```

```
        if tabuleiro:isCampoVazio(3,1) then
                print ("campo 3, 1 vazio - b")
                if                  tabuleiro:isPreenchido(3,2,BOLA)                and
tabuleiro:isPreenchido(3,3,BOLA) then
                        perde.jogada = 7;
                        perde.XIS = true;
                elseif              tabuleiro:isPreenchido(1,3,BOLA)                and
tabuleiro:isPreenchido(2,2,BOLA) then
                        perde.jogada = 7;
                        perde.XIS = true;
                elseif              tabuleiro:isPreenchido(1,1,BOLA)                and
tabuleiro:isPreenchido(2,1,BOLA) then
                        perde.jogada = 7;
                        perde.XIS = true;
                end
        end

        if tabuleiro:isCampoVazio(3,2) then
                print ("campo 3, 2 vazio - b")
                if                  tabuleiro:isPreenchido(3,3,BOLA)                and
tabuleiro:isPreenchido(3,1,BOLA) then
                        perde.jogada = 8;
                        perde.XIS = true;
                elseif              tabuleiro:isPreenchido(1,2,BOLA)                and
tabuleiro:isPreenchido(2,2,BOLA) then
                        perde.jogada = 8;
                        perde.XIS = true;
                end
        end

        if tabuleiro:isCampoVazio(3,3) then
                print ("campo 3, 3 vazio - b")
                if                  tabuleiro:isPreenchido(1,1,BOLA)                and
tabuleiro:isPreenchido(2,2,BOLA) then
                        perde.jogada = 9;
                        perde.XIS = true;
                elseif              tabuleiro:isPreenchido(3,1,BOLA)                and
tabuleiro:isPreenchido(3,2,BOLA) then
                        perde.jogada = 9;
                        perde.XIS = true;
                elseif              tabuleiro:isPreenchido(1,3,BOLA)                and
tabuleiro:isPreenchido(2,3,BOLA) then
                        perde.jogada = 9;
                        perde.XIS = true;
                end
        end
```

```
	return perde

end

function Intelligence:escolherJogada(nivel,tabuleiro)
	local jogada = {}

	--jogada para o nivel facil
	if nivel == 1 then
		jogada.x = math.random(3)
		jogada.y = math.random(3)
		return jogada
	--jogada para o nivel dificil
	elseif nivel == 2 then
		--jogada.x = 0
		--jogada.y = 0

		--verifica de XIS (computador) vencera ou perdera em uma proxima jogada
		xVence = self:verificaVence(tabuleiro)
		if not xVence.XIS then
			xPerde = self:verificaPerde(tabuleiro)
		end

		print("Xvence ou Xperde")
		print(xVence.XIS,xPerde.XIS)
		-- se XIS vencera ou perdera, entaum essa devera ser a jogada
		if xVence.XIS or xPerde.XIS then -- A "INTELIGÊNCIA" ARTIFICIAL MORA AQUI!!!
			print("entrou no if para jogar com inteligencia")
			print(xVence.jogada,xPerde.jogada)
			if xVence.jogada == 1 or xPerde.jogada == 1 then
				print("entrou na jogada 1")
				jogada.x = 1
				jogada.y = 1
			elseif xVence.jogada == 2 or xPerde.jogada == 2 then
				print("entrou na jogada 2")
				jogada.x = 1
				jogada.y = 2
			elseif xVence.jogada == 3 or xPerde.jogada == 3 then
				print("entrou na jogada 3")
				jogada.x = 1
				jogada.y = 3
			elseif xVence.jogada == 4 or xPerde.jogada == 4 then
				print("entrou na jogada 4")
				jogada.x = 2
				jogada.y = 1
			elseif xVence.jogada == 5 or xPerde.jogada == 5 then
```

```
                print("entrou na jogada 5")
                jogada.x = 2
                jogada.y = 2
            elseif xVence.jogada == 6 or xPerde.jogada == 6 then
                print("entrou na jogada 6")
                jogada.x = 2
                jogada.y = 3
            elseif xVence.jogada  == 7 or xPerde.jogada == 7 then
                print("entrou na jogada 7")
                jogada.x = 3
                jogada.y = 1
            elseif xVence.jogada == 8 or xPerde.jogada == 8 then
                print("entrou na jogada 8")
                jogada.x = 3
                jogada.y = 2
            elseif xVence.jogada == 9 or xPerde.jogada == 9 then
                print("entrou na jogada 9")
                jogada.x = 3
                jogada.y = 3
            end


-- para quando entrar de novo serem nil e false
            xVence.XIS = false
            xPerde.XIS = false
            xVence.jogada = nil
            xPerde.jogada = nil
            print("jogada intelligence:")
            print(jogada.x,jogada.y)
        -- se nao, escolhera uma jogada aleatoriamente
        else

            x = math.random(3)
            y = math.random(3)
            while not(tabuleiro:isCampoVazio(x,y)) do
                print("jogada loop aleatorio intelligence")
                x = math.random(1,3)
                y = math.random(1,3)
                print(x,y)
            end
            jogada.x = x
            jogada.y = y
            print("jogada aleatorio - intelligence")
            print(jogada.x,jogada.y)
        end
        print("jogada a ser feita:")
        print(jogada.x,jogada.y)
        -- depois de ter definido a jogada retorna as posicoes corretas
```

```
                return jogada
        end
end
```

Obs.: Os demais aplicativos e códigos fonte estão presentes na mídia externa que acompanha o projeto

5 Conclusões

A implantação da TV Digital no Brasil pode-se perceber que não houve o reconhecimento do princípio da economicidade em relação à escolha feita, haja vista que nem o padrão adotado foi o economicamente mais viável e, principalmente, não prevaleceram os direitos resguardados pela Carta Magna de 1988, diante da imposição do decreto 5820/06, que em muitos pontos entra em contradição com os princípios constitucionais.

Sabe-se que os interesses dos que mantém a televisão em termos econômico-mercadológicos sempre acabaram se sobrepondo aos interesses da população em geral. Entretanto, esta é a chance para a e democratização do sistema televisivo, haja vista abrir-se, com a digitalização, a possibilidade de surgirem novos paradigmas midiáticos. Apesar de ainda existirem ranços impositivos, percebe-se que hoje há uma consciência social muito mais pertinente que em outras épocas sobre o direito à comunicação. Há vários fóruns, centros de discussão e organizações não-governamentais que buscam caminhos para a regulamentação deste direito, apontam alternativas de diálogos e participação e procuram alertar a sociedade das suas prerrogativas, no tocante ao acesso a uma informação mais honesta, mais dialógica e mais interativa. As discussões estão tomando forma e, certamente, vão amadurecer as profícuas ideias que poderão organizar o admirável mundo novo digitalizado.

6 Referências Bibliográficas

ABNT NBR 15601 (2013) – Associação Brasileira de Normas Técnicas, “Televisão digital terrestre – Sistema de transmissão” Sistema Brasileiro de TV Digital Terrestre, NBR 15601.

ABNT NBR 15602-1 (2007) – Associação Brasileira de Normas Técnicas, “Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação – Parte 1: Codificação de vídeo”, Sistema Brasileiro de TV Digital Terrestre, NBR 15602-1.

ABNT NBR 15602-2 (2007) – Associação Brasileira de Normas Técnicas, “Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação – Parte 2: Codificação de vídeo”, Sistema Brasileiro de TV Digital Terrestre, NBR 15602-2.

ABNT NBR 15606-2 (2007) – Associação Brasileira de Normal Técnicas, “Televisão digital terrestre – Codificação de dados e especificação de transmissão para radiodifusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações”, Sistema Brasileiro de TV Digital Terrestre, NBR 15606

ABRÃO, I.C; BARRÉRE, E. e TEIXEIRA, C.A. Ambiente para Intercâmbio de Objetos Multimídia: Aspectos Computacionais.

BARBOSA, S. D. J. e SOARES, L. F. G. Tutorial: TV Digital Interativa se faz com Ginga: fundamentos, padrões, autoria declarativa e usabilidade.

CAMOLESI, A.R. Uma metodologia para o Design de Serviços de TV-Interativa. Dissertação de Mestrado, PPG-CC, UFSCar, 2000.F

GINGA. Middleware Aberto do Sistema Nipo-Brasileiro de TV Digital (ISDB-TB). disponível em: http://www.ginga.org.br/pt-br, acesso dezembro de 2011

ITU-R BT.601-4(1994) International Communication Union “Encoding Parameters of Digital Television for Studios”, Recommendation BR.601-4, BR Series Volume, Geneva

SOARES, L. F. G.; BARBOSA, S. D. J. Programando em NCL – Desenvolvimento de aplicações para middleware ginga, TV Digital e web. 1ª edição. Rio de Janeiro: Campus, 2009. 360p.

W3C REC-xhtml1-20130801(2013) Word Wide Web Consortium, “XHTML™ 1.0 The Extensible Hipertext Markup Language”, W3C Recommendation xhtml1-2013801