

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA TV DIGITAL COM DIVULGAÇÃO DE INFORMAÇÕES SOBRE O TRANSPORTE PÚBLICO

Carlos Marques¹, ElyssandroPiffer¹, Izabel C. Mioranza¹, Leandro C. de Lima¹

¹Instituto de Ciências Exatas, Agrárias, Tecnológicas e Geociências – Universidade Paranaense (UNIPAR) – Unidade Umuarama – Sede

Pç. Mascarenhas de Moraes, 4282 - Zona III - 87502-210 - Umuarama – PR – Brasil

{cmarques, piffer, izabel, leandro}@unipar.br

Abstract. This file describes the main aspects involved in developing applications for SBTVD (Sistema Brasileiro de Televisão Digital), using as an object of study the development of an application that aims to create opportunities to the communities served by public transport services the information about it. The proposed application was implemented using the NCL declarative language, combined with procedural language LUA, which have their bases covered throughout the text, as well as development and test environment. The developed application provides information on routes, lines, schedules and points of access to public transportation in a city or region.

Resumo. Este artigo descreve os principais aspectos envolvidos no desenvolvimento de aplicações para SBTVD (Sistema Brasileiro de Televisão Digital), utilizando como objeto de estudo, a elaboração de um aplicativo que visa oportunizar as comunidades atendidas por serviços de transporte público informações sobre o mesmo. A aplicação proposta foi implementada utilizando a linguagem declarativa NCL, aliada a linguagem procedural LUA, as quais têm seus fundamentos abordados no decorrer do texto, bem como o ambiente de desenvolvimento e testes. A aplicação desenvolvida disponibiliza informações sobre rotas, linhas, horários e pontos de acesso ao transporte público de uma cidade ou região.

1. INTRODUÇÃO

Após o surgimento da televisão (TV) analógica em 1924, várias novas tecnologias foram sendo implementadas agregando valor a este importante veículo de comunicação e informação para a população mundial.

A transmissão em cores, o som estéreo, o *close caption*(legenda oculta) e outras implementações agregaram valor a TV ao longo do tempo, porém algumas outras novidades como a alta definição de imagem e o som com maior qualidade ficaram limitadas ao meio de transmissão analógico, sendo que este impossibilitava maiores avanços por ser uma tecnologia com poucas possibilidades de ser incrementada.

Para resolver esta situação, e principalmente com o intuito de oferecer um serviço de maior qualidade, vários estudos foram iniciados, culminando com a proposta da TV digital, que como o próprio nome sugere, tem como base de funcionamento a transmissão digital do sinal desde a geradora/transmissora até cada aparelho receptor. Com este tipo de transmissão, que teve seus estudos iniciados na década de 1980, foi possível agregar mais qualidade de som e imagem, além de outras funcionalidades.

O Brasil ficou a margem desta tecnologia por certo tempo, ao se considerar que em 1998, cada qual com o seu respectivo sistema, o Reino Unido e os Estados Unidos iniciavam este tipo de transmissão e, o Japão com sistema diferente dos demais, começou a operar este sistema comercialmente em 2003. A partir de 2 de dezembro de 2007 o Brasil adotou esta tecnologia, com a primeira transmissão sendo feita para a cidade de São Paulo, e gradativamente está implantando a mesma em todo seu território.

O padrão adotado pelo Brasil é diferente de todos os outros previamente existentes, sendo que a proposta desde o início, era de se criar um padrão tecnologicamente avançado e que pudesse ser aliado nas políticas públicas de disseminação do conhecimento, devendo portanto ser uma opção de baixo custo e que não envolvesse muitos *royalties*, deixando assim os equipamentos, principalmente os receptores, com o menor preço possível.

Para tornar isto possível, porém sem ter que desenvolver uma tecnologia a partir do início, o que levaria um tempo razoável, e como dito anteriormente, já havia uma defasagem em relação a certas regiões do mundo, optou-se por utilizar um dos padrões previamente existentes como base, agregando ao mesmo, novas características ou melhorando aquelas já existentes. O padrão japonês ISDB-T (*Integrated Services Digital Broadcasting Terrestrial*) foi o escolhido, após vários estudos e discussões, para ser o ponto de partida para o SBTVD (Sistema Brasileiro de Televisão Digital) [DTV2014]. O SBTVD está se consolidando não só no território brasileiro, mas como uma referência para outros países, sendo que alguns deles já o adotaram oficialmente, como: Peru, Argentina, Chile, Venezuela, Equador, Costa Rica, Paraguai, Filipinas, Bolívia, Nicarágua e Uruguai.

Portanto a TV digital brasileira apresenta atualmente tecnologia avançada e que propicia várias vantagens para a população. A vantagem mais perceptível é quanto à qualidade da programação recebida, tanto pela imagem em alta definição, como pelo som com maior fidelidade, eliminando imagens fantasmas e ruídos no som. Entretanto existem outros aspectos que ainda não são tão perceptíveis à população em geral, mas que podem agregar mais valor, como a possibilidade de obter informações, realizar compras, participar ativamente de programas de televisão através de enquetes, entre outros. Estes recursos compõem a interatividade, que também se apresenta nos demais sistemas de TV digital do mundo.

Para que todos estes recursos estejam disponíveis para o telespectador, é necessário que o aparelho receptor esteja apto a decodificar o sinal transmitido de forma a processar a

imagem, o som e a interatividade. Os novos aparelhos receptores já vêm com este recurso embutido, não sendo necessário qualquer tipo de adaptação, porém para os aparelhos antigos, deve-se adquirir um conversor, conhecido por *top box*, que irá desempenhar estas funções.

A interatividade no SBTVD é possível pela implementação do *middleware Ginga*. Este componente é responsável por fazer a ponte entre o equipamento e a aplicação desenvolvida e, foi desenvolvido em solo brasileiro, mais especificamente pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ) e a Universidade Federal de Paraíba (UFPB). Para desenvolver uma aplicação que permita a interatividade, pode-se optar basicamente entre dois ambientes: Ginga-NCL e Ginga-J, sendo que o primeiro utiliza a linguagem de contexto aninhado (NCL) e LUA, já o segundo emprega a linguagem Java [GINGA2014].

O recurso de interatividade atualmente é subutilizado, sendo que as possibilidades que o mesmo permite são desmedidas e, para a população de um país com as características do Brasil, certamente poderiam ser mais exploradas. Portanto, para colaborar para esta disseminação de informações úteis para a população, foi proposto o desenvolvimento de uma aplicação utilizando NCL e LUA para a divulgação de dados importantes sobre o transporte público de uma região.

Os dados a serem divulgados são as linhas, os horários, os itinerários e os pontos de acesso ao transporte público de uma determinada região. Estes dados serão transmitidos para o aparelho receptor, juntamente com o aplicativo que possibilitará que o usuário escolha a pesquisa que desejar. Cabe ressaltar que esta transmissão é efetuada juntamente com a transmissão da grade normal da emissora de TV, pois o sinal enviado é composto por vídeo, áudio e um canal específico para os dados e aplicativos. Esta junção dos diferentes fluxos é efetuada na transmissora através de aparelho específico e, ao chegar no receptor, é separado e destinado a cada módulo de processamento.

Com esta aplicação pretende-se, além do aspecto da pesquisa de novas tecnologias, a consolidação de conhecimentos, mas principalmente a prestação de serviços à comunidade, sendo estes os papéis a serem desenvolvidos pela academia.

2. AMBIENTE NCLUA

Para o desenvolvimento da aplicação proposta, que envolve interação com o usuário no ambiente de TV Digital utilizando o Ginga, a opção escolhida foi o uso da linguagem NCL, que foi desenvolvida pelo laboratório de telemídia do departamento de informática da PUC-RIO (Pontifícia Universidade Católica do Rio de Janeiro). Trata-se de uma linguagem declarativa baseada em XML (*eXtensible Markup Language*) a qual possui um modelo conceitual de dados, denominado por NCM (*Nested Context Model*). O modelo propõe representar conceitos estruturais de dados, eventos e relacionamentos entre os dados [GINGA2014].

A NCL faz utilização do modelo conceitual de dados para elaboração de documentos hipermídia, porém apresenta algumas desvantagens onde, por ser uma linguagem declarativa não dispõe de controle total das informações processadas na aplicação, portanto apesar dos recursos que fornece, em muitos casos, ela não é por si só suficiente para atender a todas as necessidades no desenvolvimento de aplicações mais complexas ou que exijam uma maior interação com o usuário. Necessitando assim de uma junção com uma linguagem procedural como a LUA, linguagem esta desenvolvida nos laboratórios Tecgraf (Desenvolvimento de Software Técnico-Científico da PUC-Rio) no início dos anos 1990. LUA foi totalmente projetada para trabalhar estendendo outras aplicações, por isso é muito

raro encontrar aplicações totalmente desenvolvidas nesta linguagem, pois trabalha com scripts, e seu próprio nome remete a uma linguagem “satélite”, apresentando muita utilidade, mas nunca com a função principal.“[...] LUA é uma linguagem de fácil aprendizado, que combina sintaxe procedural com declarativa, com poucos comandos primitivos [...]” [SOARES e BARBOSA2009].

As vantagens citadas pelos próprios desenvolvedores sobre LUA é a robustez, rapidez e principalmente ser código livre. Portanto esta linguagem combina perfeitamente com as necessidades que a NCL tem para produzir as aplicações para a TV Digital. A portabilidade talvez seja um dos pontos mais importantes para a utilização de LUA como linguagem *script*, pois o cenário de utilização do *middleware* Ginga pode ser o mais variável possível, exigindo máxima flexibilidade de ambos.

3. AMBIENTE DE DESENVOLVIMENTO

Para a televisão digital, existem diversas ferramentas que realizam a codificação e execução das aplicações interativas, dentre elas, está o IDE (*Integrated Development Environment*) conhecida como Eclipse. O Eclipse apresenta diversas funcionalidades que facilitam a codificação e execução de aplicações para a televisão digital [ECLIPSE2014].

Também foi necessário a utilização do *plug-in* NCL Eclipse, que auxilia e agiliza o desenvolvimento de aplicações escritas na linguagem NCL. Entre suas principais características, está o fato que o NCL Eclipse pode sugerir o código a ser escrito, marcar erros na aplicação e executar as aplicações [NCL ECLIPSE2014].

Para editar scripts na linguagem LUA, foi utilizado o *plug-in* denominado LUA Eclipse. Com este *plug-in* foi possível editar *scripts* com realce de sintaxe, completar código automaticamente, realizar a verificação de erros de compilação, agrupar códigos e comentários e executar os *scripts* utilizando de um interpretador devidamente configurado. [LUA ECLIPSE2014].

O ambiente de simulação de um aparelho receptor de televisão digital também foi necessário para testar as funções de interatividade ao executar a aplicação. Para isto foi utilizada uma máquina virtual com Ginga-NCL Virtual STB [VIRTUALSTB 2014] emulada através do software *VMware Player* [VMWARE2014].

Para acesso a máquina virtual contendo o Ginga foi necessário a utilização de um cliente FTP (*File Transfer Protocol*). Para tanto foi utilizado o *WinSCP* que é um programa bastante utilizado para realizar esta função [WINSCP2014].

4. APLICAÇÃO DESENVOLVIDA

É senso comum que o transporte público no Brasil apresenta-se de forma inconsistente e deficitário, com a qualidade de serviço muito contestada. Além de todos os outros problemas apresentados como o número reduzido de veículos, o péssimo estado de conservação dos mesmos, a quantidade de rotas e horários inapropriados para a necessidade da população, a falta de informação é outro fator crítico, pois os usuários muitas vezes não sabem quais são as linhas ofertadas em determinada região, quais os horários que circulam veículos nestas linhas, quais as rotas percorridas pelas linhas e, quais os pontos de embarque e desembarque.

Objetivando minimizar este problema de informação sobre o transporte público, identificou-se a necessidade de se implementar um software utilizando a televisão digital

como plataforma, pois a televisão é indiscutivelmente o veículo de comunicação mais utilizado na América Latina, bem como no Brasil onde chega a 96% o índice de audiência segundo o IBOPE. [IBOPE 2015].

Cabe ressaltar que o uso desta tecnologia possibilita também o acesso móvel, uma vez que os fabricantes de celular (*smartphones*) vêm inserindo a tecnologia da televisão digital, juntamente com o Ginga em seus aparelhos, logo independentemente do local, a possibilidade de se ter um aparelho que disponibilize o acesso a estes dados tem se tornado cada vez mais presente, uma vez que este tipo de dispositivo tem apresentado altos índices de venda, atingindo também a população de menor renda.

A premissa básica desta aplicação é o fato de que o principal público a ser favorecido apresenta um índice maior de distanciamento da tecnologia, portanto esta deveria ser simples e apenas com as principais informações, facilitando assim o acesso. Portanto quanto menos passos o usuário precisasse efetuar para obter os dados necessários, melhor seria sua experiência com a nova tecnologia e, conseqüentemente, maior as chances do aplicativo atingir os seus objetivos. Uma vez que a aplicação está disponível para acesso, o aparelho receptor demonstra esta disponibilidade através de uma imagem no canto esquerdo da tela, juntamente com o programa televisivo que estiver sendo transmitido naquele momento. Neste momento o usuário deve pressionar o botão correspondente no controle do aparelho receptor para acessar a aplicação.

Quando a aplicação começar a ser executada, a imagem do programa de televisão que está sendo transmitido será redimensionada, possibilitando assim que o mesmo fique ainda disponível para o usuário, porém em uma tela de menor dimensão, localizada no canto superior direito, conforme a Figura 1.

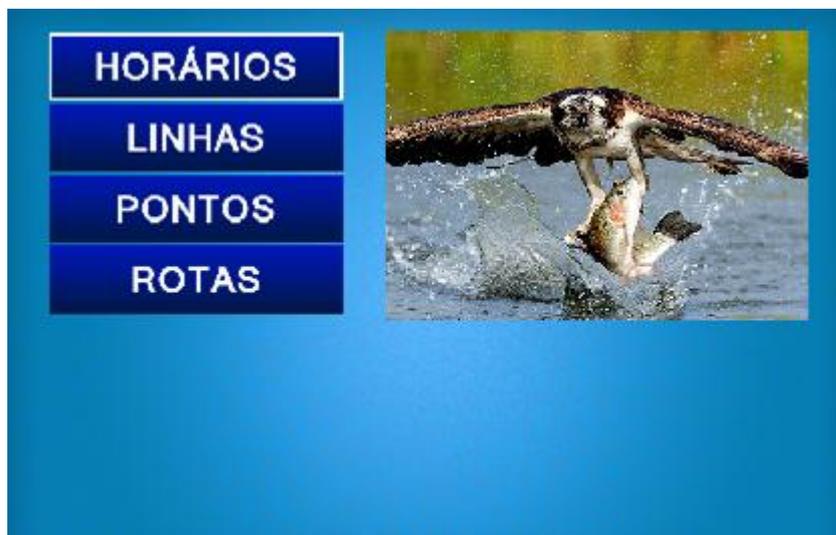


Figura 1. Aplicação iniciada

Neste momento, conforme a figura 1, serão disponibilizadas para o usuário quatro opções: Horários, Linhas, Pontos e Rotas. Estas opções estarão em forma de botões onde o usuário deverá selecionar a opção desejada através das setas do controle remoto.

Após a seleção da opção desejada, o usuário deverá pressionar a tecla OK no controle remoto e então as informações referentes a aquela opção serão mostradas na tela em uma área

específica, logo abaixo dos botões e da imagem do programa televisivo. Este procedimento pode ser observado na figura 2.



Figura 2 – Seleção da opção desejada

Quando o usuário seleciona a opção de horários, são listados todos os horários de todas as linhas. Esta lista é mostrada em ordem do nome da linha e por horário, onde o usuário pode navegar pela mesma através das teclas de controle do aparelho. Ainda nesta lista os dados apresentados são o código da linha, o nome da linha e os respectivos horários.

Se a opção selecionada pelo usuário for a de linhas, são apresentadas todas as linhas em ordem alfabética, constando o código e o nome da linha, conforme pode ser observado na Figura 2.

Caso o usuário selecione a opção de pontos, são mostrados todos os pontos ordenados por endereço.

Para a opção de rotas, são listadas todas as linhas em ordem alfabética, mostrando as respectivas rotas, indicando todos os pontos por onde a linha passa ao longo do seu roteiro.

Todas estas informações são desvinculadas do código da aplicação, sendo que a mesma deve consultar arquivos em formato texto, mais especificamente no formato CSV (*Comma Separated Values*), com separação dos campos pelo caractere“;” (ponto e vírgula). Estes arquivos são transmitidos juntamente com a aplicação através do sistema de carretel disponibilizado por aparelhos específicos instalados nas estações transmissoras de televisão digital que são responsáveis por encadear as aplicações e seus respectivos dados. Os arquivos a serem disponibilizados junto da aplicação são os arquivos de linhas, rotas, horários e pontos.

Por esta separação da aplicação em relação aos arquivos que são acessados para disponibilizar o conteúdo para o usuário, pode-se implementar funcionalidades adicionais como, por exemplo, disponibilizar informações de várias cidades de uma região metropolitana, onde o usuário pode escolher de qual cidade deseja as informações logo no início da aplicação, possibilitando o acesso a arquivos específicos da cidade desejada. Desta forma uma mesma emissora poderia ter uma área de abrangência de cobertura maior no que se refere às informações prestadas pela aplicação.

Outro ponto a se destacar é que com a adoção de arquivos separados da aplicação, as empresas prestadoras de serviços de transporte público podem a qualquer momento, efetuar as

devidas atualizações nos arquivos e os enviar para a emissora de TV que apenas substitui os arquivos antigos pelos novos, sem a necessidade de intervenção de um programador, tornando-o assim um sistema de fácil manutenção.

4.1 Estrutura do projeto da aplicação

Através do *plug-in* NCL Eclipse foi gerado automaticamente o arquivo *“defaultConnBase.ncl”*. Este arquivo apresenta-se em formato XML e possui as definições padrões para a aplicação. Foi criado também o arquivo *“principal.ncl”*, responsável por toda parte de visualização e controle personalizada da mesma, inclusive com as ligações necessárias com LUA.

Com a finalidade de estruturar melhor o projeto, foram criadas pastas, onde na pasta principal ficam os arquivos mencionados acima, enquanto que os demais arquivos ficam na pasta *“media”*, organizados nas subpastas *“audio”*, *“img”*, *“lua”* e *“video”*.

A pasta *“img”* contém as imagens a serem utilizadas na aplicação. Estas imagens podem ser as utilizadas nos botões ou em outras partes da interface gráfica. Quanto aos arquivos texto em formato CSV e os arquivos em LUA, ficaram na pasta *“lua”*, sendo que o arquivo de vídeo, utilizado como exemplo e simulação do programa televisivo a ser apresentado ficou na pasta *“video”*.

4.2 Sobre o arquivo NCL

O arquivo *“principal.ncl”* é o responsável por estruturar a aplicação do ponto de vista da visualização, determinando posições onde os componentes serão apresentados, bem como controlar eventos e associar alguns destes eventos aos arquivos LUA que processam requisições do usuário lendo os arquivos CSV e mostrando os conteúdos desejados.

Inicialmente o arquivo NCL determina as regiões (*regions*), especificando assim, a localização e o tamanho que cada componente irá ocupar na tela.

Na sessão *head* do arquivo, inicia-se com as definições das regiões, onde pode-se observar que são declaradas várias destas, inclusive a região específica do menu de opções para o usuário *“RgMenu”* com suas respectivas regiões. As regiões são compostas pelo *“id”* que estabelece um nome identificador que pode ser utilizado em outras partes do arquivo, quando necessário. Também são definidas características como posicionamento na tela e o tamanho de cada região através dos atributos *left*, *top*, *width* e *height*. Outro atributo definido é o *“zIndex”* que determina em qual camada (*“layer”*) a região será disposta na tela, onde valores maiores indicam que a região deve estar na frente da tela em relação a regiões com valores inferiores, conforme pode ser observado na figura 3.

```
<regionBase id="RgPrincipal">
<region id="RgVideo" left="0.00%" top="0.00%" width="100.00%" height="100.00%" zIndex="5"/>
<region id="RgFundo" left="0.00%" top="0.00%" width="100.00%" height="100.00%" zIndex="1"/>
<region id="RgMenu" left="5.00%" top="5.00%" width="35.00%" height="50.00%" zIndex="3">
  <region id="RgMenu1" height="24.00%" zIndex="3"/>
  <region id="RgMenu2" height="24.00%" top="25.00%" zIndex="3"/>
  <region id="RgMenu3" height="24.00%" top="50.00%" zIndex="3"/>
  <region id="RgMenu4" height="24.00%" top="75.00%" zIndex="3"/>
</region>
<region id="RgRodape" left="0.00%" top="95.00%" width="100.00%" height="5.00%" zIndex="4"/>
<region id="RgHorariosLua" left="2.5%" right="2.5%" top="60%" width="95%" height="33%" zIndex="6"/>
<region id="RgLinhasLua" left="2.5%" right="2.5%" top="60%" width="95%" height="33%" zIndex="6"/>
<region id="RgPontosLua" left="2.5%" right="2.5%" top="60%" width="95%" height="33%" zIndex="6"/>
<region id="RgRotasLua" left="2.5%" right="2.5%" top="60%" width="95%" height="33%" zIndex="6"/>
<region id="RgContainer" left="2.5%" right="2.5%" top="60%" width="95%" height="33%" zIndex="5"/>
</regionBase>
```

Figura 3 – Definição das regiões

A Figura 4 apresenta a definição dos descritores que são os responsáveis pela configuração de como os objetos de mídia devem ser apresentados. Cada descritor tem seu “*id*” que estabelece um nome identificador e a região a qual o mesmo está associado, através do atributo “*region*”. Pode-se notar que os descritores ligados as regiões dos itens de menu apresentam atributos específicos que determinam a ordem com que o foco será atribuído conferindo assim as regras de navegabilidade que a aplicação obedece durante a interação do usuário com o controle remoto do aparelho. O atributo *focusIndex* determina o identificador do foco, enquanto que *moveUp* e *moveDown* estabelecem para quais itens do menu o foco deve ser deslocado quando o usuário pressionar os comandos, respectivamente, para subir ou descer. Quanto ao atributo *focusBorderColor*, apenas define a cor da borda do componente, quando este estiver com o foco.

```

<descriptorBase id="DescPrincipal">
<descriptor id="DescVideo" region="RgVideo"/>
<descriptor id="DescFundo" region="RgFundo"/>
<descriptor id="DescRodape" region="RgRodape"/>
<descriptor id="DescMenu" region="RgMenu"/>
<descriptor id="DescMenu1" region="RgMenu1" focusIndex="1" moveUp="4" moveDown="2" focusBorderColor="white"/>
<descriptor id="DescMenu2" region="RgMenu2" focusIndex="2" moveUp="1" moveDown="3" focusBorderColor="white"/>
<descriptor id="DescMenu3" region="RgMenu3" focusIndex="3" moveUp="2" moveDown="4" focusBorderColor="white"/>
<descriptor id="DescMenu4" region="RgMenu4" focusIndex="4" moveUp="3" moveDown="1" focusBorderColor="white"/>
<descriptor id="DescHorariosLua" region="RgHorariosLua"/>
<descriptor id="DescLinhasLua" region="RgLinhasLua"/>
<descriptor id="DescPontosLua" region="RgPontosLua"/>
<descriptor id="DescRotasLua" region="RgRotasLua"/>
<descriptor id="DescContainer" region="RgContainer"/>
</descriptorBase>

```

Figura 4 – Definição dos descritores

Em NCL as ações a serem tomadas são determinadas a partir de uma causa, onde cada ação é precedida de outra. Deve haver uma condição satisfeita antes de se disparar uma ação. Logo o sincronismo da aplicação ao longo do tempo e do espaço segue uma lógica, a lógica causal.

A Figura 5 apresenta a definição dos conectores que são os responsáveis pela ligação de uma condição a uma determinada ação. Inicialmente define-se como definições básicas as constantes no arquivo *defaultConnBase.ncl*, a quem é atribuído um *alias* (apelido) de *conn*. Na sequência são definidos adicionalmente dois conectores: *onSelectionStart* e *onBeginStart*, onde ambos definem como ação *start* (iniciar), porém em condições diferentes: *onSelection* (ao selecionar) e *onBegin* (ao iniciar), respectivamente.

```

<connectorBase id="connBase">
<importBase alias="conn" documentURI="defaultConnBase.ncl"/>
<causalConnector id="onSelectionStart">
<simpleCondition role="onSelection"/>
<simpleAction role="start"/>
</causalConnector>
<causalConnector id="onBeginStart">
<simpleCondition role="onBegin"/>
<simpleAction role="start"/>
</causalConnector>
</connectorBase>

```

Figura 5 – Definição dos conectores

Já na sessão *body* do arquivo NCL, há a definição das mídias e a associação destas aos seus respectivos descritores, conforme a figura 6 apresenta. Cada mídia tem seu “*id*” que estabelece um nome identificador. O atributo *src* determina a localização da mídia e o *descriptor* vincula a mesma a um descritor.

As mídias *horariosLua*, *linhasLua*, *pontosLua* e *rotasLua* são associadas aos seus respectivos arquivos que contêm os códigos em linguagem LUA a serem executados quando necessário.

```
<media id="video1" src="media/video/video1.mp4" descriptor="DescVideo">
<propertyname="bounds"/>
</media>
<media id="fundo" src="media/img/fundo.jpg" descriptor="DescFundo"/>
<media id="rodape" src="media/img/rodape.jpg" descriptor="DescRodape"/>
<media id="horarios" src="media/img/horarios.jpg" descriptor="DescMenu1"/>
<media id="linhas" src="media/img/linhas.jpg" descriptor="DescMenu2"/>
<media id="pontos" src="media/img/pontos.jpg" descriptor="DescMenu3"/>
<media id="rotas" src="media/img/rotas.jpg" descriptor="DescMenu4"/>
<media id="horariosLua" src="media/lua/testehor.lua" descriptor="DescHorariosLua"/>
<media id="linhasLua" src="media/lua/linhas.lua" descriptor="DescLinhasLua"/>
<media id="pontosLua" src="media/lua/pontos.lua" descriptor="DescPontosLua"/>
<media id="rotasLua" src="media/lua/rotas.lua" descriptor="DescRotasLua"/>
<media id="container" src="media/img/container.png" descriptor="DescContainer"/>
```

Figura 6 – Definição das mídias

Na sequência, foi definida a porta de entrada da aplicação, ou seja, por onde a aplicação deve iniciar suas atividades. Para tanto foi utilizada a linha de código, `<port id="p0" component="video1"/>`, definindo como esta porta o *video1*.

Finalizando, foram definidos os *links* que determinam quais ações a serem tomadas quando cada evento (causa) acontecer. Este artigo absteve-se de apresentar todo o código desta parte, uma vez que o mesmo apresenta-se relativamente longo, porém apresenta-se um exemplo de como foi elaborado.

A figura 7 demonstra um exemplo de *link*, que trata do pressionamento da tecla *Menu*, por parte do usuário. O atributo *id* estabelece um identificador para o *link*, sendo que o atributo *xconnector* liga o mesmo ao conector `conn#onKeySelectionSet_varStart`, conector este definido no arquivo *defaultConnBase.ncl*, que tem como *alias* (apelido) *conn*. Portanto, ao selecionar a referida tecla, uma série de ações será desencadeada, onde os componentes *fundo*, *horários*, *linhas*, *pontos*, *rotas* e *rodape* serão iniciados, ou seja, mostrados na tela e, o componente *video1* será redimensionado para que permita que os demais componentes possam ser melhor visualizados. Desta forma a tela do aplicativo será apresentada para o usuário com o menu para que o mesmo possa efetuar suas escolhas sobre quais informações deseja.

```
<link id="IMenu" xconnector="conn#onKeySelectionSet_varStart">
<bind role="onSelection" component="video1">
<bindParam name="keyCode" value="MENU"/>
</bind>
<bind role="start" component="fundo"/>
<bind role="start" component="horarios"/>
<bind role="start" component="linhas"/>
<bind role="start" component="pontos"/>
<bind role="start" component="rotas"/>
<bind role="start" component="rodape"/>
<bind role="set" component="video1" interface="bounds">
<bindParamname="var" value="45%,5%,50%,50%"/>
</bind>
</link>
```

Figura 7 – Exemplo de link

4.3 Sobre os arquivos LUA

Para cada informação específica que o usuário desejar, o aplicativo dispõe de um arquivo LUA referente à mesma, portanto existem os arquivos *horarios.lua*, *linhas.lua*, *pontos.lua* e *rotas.lua*. Cada um destes arquivos contém o código em linguagem LUA que acessa seu respectivo arquivo CSV para dele extrair os dados necessários para apresentar ao usuário.

Este artigo deixar de apresentar todos os códigos em razão do tamanho dos mesmos, porém toma o arquivo *linhas.lua* como exemplo para explicar como o mesmo funciona, uma vez que os demais seguem a mesma linha de raciocínio.

A figura 8 apresenta o início do código LUA, que será executado assim que o NCL o requisitar. Nesta parte são declaradas e inicializadas as variáveis necessárias para o processamento, bem como o arquivo *linhas.txt* é aberto para leitura e todos os seus dados são colocados no vetor *dados*, através da estrutura de repetição *for*. Ao término do código o arquivo é fechado.

A linha *event.register(LuaNCL)* é responsável por vincular a função *LuaNCL*, descrita a seguir, como a função a ser executada quando da chamada do código NCL.

```
local dx, dy = canvas:attrSize()
local y = 0
local x = 0
nomeArquivo = "linhas.txt"
arquivo = assert(io.open(nomeArquivo, "r"), "Arquivo não pode ser criado")
arquivo:seek("set")
linhas = {}
dados = {
  nome = {}, colunas = {}
}
line = 0
i = 1
counter = 1
count = 0

--Processamento das informações encontradas no arquivo
for line in io.lines("linhas.txt") do
  linhas[counter] = arquivo:read('*line')
  tamanho = string.len(linhas[counter])
  pos = string.find(linhas[counter], ';')
  dados.nome[counter] = string.sub(linhas[counter],i, pos-1)
  io.write(dados.nome[counter] .. '\n')
  dados.colunas[counter] = string.sub(linhas[counter],pos+1, tamanho-1)
  counter = counter + 1
end
event.register(LuaNCL)
arquivo:flush()
io.close(arquivo)
```

Figura 8 – Inicialização do arquivo LUA

Após a inicialização do código LUA, existe a função que será a função propriamente dita a ser executada pela chamada recebida do código em NCL. Esta função é apresentada na figura 9. Esta função inicialmente define a configuração de como serão as cores, o tamanho e o tipo dos caracteres a serem apresentados na tela. Na sequência os dados presentes no vetor *dados* são impressos.

Cabe ressaltar que após impressos os dados, deve-se gerar um evento de aviso ao NCL de que os dados estão prontos para serem realmente impressos na tela para o usuário. Esta ação se dá pelo uso da instrução *event.post*.

```

functionLuaNCL (evt)
ifevt.class == 'ncl' and evt.type == 'presentation' and evt.action == 'start' then
canvas:attrColor ('black')
canvas:attrFont ('arial', 16, 'bold')
canvas:drawText (30,y,"Codigo da linhas:")
canvas:drawLine(30, 20, 200, 20)
y = y + 20
forvarNome=1, #dados.nome do
canvas:drawText (30,y, dados.nome[varNome])
tamanho = canvas:measureText(dados.nome[varNome])
if x < tamanho then
x = tamanho
x = x + 20
end
    y = y + 20
canvas:flush()
event.post {
class = 'ncl',
type = 'presentation',
action = 'start'
}
end
    y = 0
canvas:drawText (300,y,"Linha:")
canvas:drawLine(300, 20, 500, 20)
y = y + 20
forvarDados=1, #dados.colunas do
canvas:drawText (300,y, dados.colunas[varDados])
y = y + 20
canvas:flush()
event.post {
class = 'ncl',
type = 'presentation',
action = 'start'
}
end
end
end
end

```

Figura 9 – Função LuaNCL

5. CONSIDERAÇÕES FINAIS

Através do desenvolvimento deste projeto pode-se perceber o potencial que a interatividade tem para um país de dimensões continentais como o Brasil, no que tange a disseminação de conhecimento e informação.

Com a popularização da TV Digital e a crescente migração do sistema de transmissão analógica para esta nova tecnologia, cada vez mais pessoas poderão se beneficiar das vantagens trazidas pela mesma.

Outro ponto que ratifica a importância de aplicações interativas na televisão digital é o fato de que, ao contrário do que ocorria anteriormente, os equipamentos atuais estão cada vez mais deixando de desempenhar funções muito específicas, abrangendo outras funções que originalmente não eram de sua competência. Esta realidade é expressa pelo termo *Convergência Digital*, que pode ser facilmente observada nos *smartphones* que desempenham várias funções, além de somente fazer e receber ligações telefônicas. Ainda usando o exemplo destes aparelhos, está cada vez mais comum observar exemplares dos mesmos com a possibilidade de acesso a televisão digital, logo a preferência da população pela televisão como forma de comunicação e diversão pode ser ainda maior, pois agora lhes é oferecida a opção de poder assistir ao seu programa favorito em qualquer lugar. Se por um lado isto é interessante do ponto de vista da comunicação e diversão, por outro pode-se agregar a isto a possibilidade da disseminação de informação de utilidade pública.

Apesar disto, ainda são poucas as aplicações desenvolvidas, principalmente com este intuito, sendo que a maioria tem ligação com a grade de programação das emissoras, sobre programas específicos, ou ainda, de cunho comercial, objetivando a venda de produtos e serviços. Toda tecnologia pode ser utilizada de forma benéfica ou maléfica, depende da forma com que se a utiliza, porém a administração pública direta ou indireta, seja por autarquias, empresas públicas, ou ainda por concessionárias de serviços públicos, podem e devem utilizar-se destas tecnologias no sentido de buscar oferecer a população um melhor serviço, sem com isto onerar demasiadamente suas planilhas de custo.

Outro fator a ser discutido é a tecnologia envolvida, onde pode-se perceber que apesar de algo novo, o uso das linguagens NCL e LUA é relativamente fácil e pode render bons resultados finais. Se por um lado NCL se apresenta de fácil manuseio e com relativa adaptabilidade quanto ao controle de mídias e eventos, LUA a complementa nos seus pontos frágeis, possibilitando um processamento mais robusto, inclusive com a manipulação de arquivos.

Durante a elaboração da aplicação, surgiu a ideia de suprimir os arquivos em formato CSV, passando da interatividade local para a interatividade através de canal de retorno, utilizando a internet. Porém esta ideia foi deixada para outro momento, uma vez que limitaria em demasia o número de aparelhos capazes de trabalhar a contento com a aplicação, e consequentemente diminuiria drasticamente a população potencialmente atendida.

A aplicação em si foi desenvolvida e testada em ambiente simulado, sendo que a intenção é de colocá-la em produção em alguma operadora, porém pela falta de equipamentos específicos nas emissoras locais para inserir a aplicação e os dados, juntamente com o conteúdo que já é transmitido (vídeo e áudio), não foi possível até o momento. Porém ainda há o contato com emissoras, que dizem ter interesse em viabilizar o projeto através da aquisição destes equipamentos num futuro próximo.

REFERÊNCIAS

- ABNT NBR 15606-2. Televisão digital terrestre – codificação de dados e especificações de transmissão para radiodifusão digital, Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Associação Brasileira de Normas Técnicas, 2008.
- BATISTA, G.B.; GOMES, F.J.L. Desenvolvimento de aplicações para TV digital interativa. Parnaíba – PI, Instituto Federal do Piauí: VI ENUCOMP, 2013.
- CARVALHO, R.; SANTOS, J.A.F. et al. Introdução às linguagens NCL e LUA: desenvolvendo aplicações interativas para TV digital. 2009. 77 p. Universidade Federal Fluminense, Niterói, 2009.
- DTV. Site oficial da televisão digital brasileira. Disponível em: <http://www.dtv.com.br>. Acesso em 21 ago. 2014.
- ECLIPSE. Eclipse - The eclipse foundation open source community website. Disponível em: <https://www.eclipse.org/home/index.php>. Acessado em 22 out. 2014.
- GINGA. Sobre o ginga. Disponível em: <http://www.ginga.org.br/pt-br/sobre>. Acesso em: 21 ago. 2014.
- GINGA-NCL. Sobre o ginga-NCL. Disponível em: <http://www.gingaNCL.org.br/pt-br/sobre>. Acesso em: 21 ago. 2014.

- IBOPE – TV aberta segue como o meio de maior penetração na América Latina. Disponível em: <http://www.ibope.com.br/pt-br/noticias/paginas/tv-aberta-segue-como-o-meio-de-maior-penetracao-na-america-latina.aspx>. Acessado em 16 set. 2015.
- LABORATÓRIO TELEMÍDIA. Nested Context Language. Disponível em: <http://www.telemidia.puc-rio.br/>. Acesso em: 20 de ago. 2014.
- LUA. A Linguagem de programação LUA. Disponível em: <http://www.LUA.org/portugues.html>. Acesso em 22 out. 2014.
- LUAECLIPSE. Um ambiente de desenvolvimento integrado para a linguagem de programação LUA. Disponível em <http://LUAeclipse.LUAforge.net/>. Acessado em 22 out. 2014.
- MELO, J.C.P.; ARAÚJO, R.M. Os módulos NCL e NCLUA do middleware Ginga para aplicação em TV digital interativa. Natal: Universidade Federal do Rio Grande do Norte, 2008.
- NCL. Visão geral. Disponível em: <http://www.NCL.org.br/pt-br/inicio/>. Acesso em: 21 ago. 2014.
- NCLECLIPSE. Introdução – NCL eclipse. Disponível em: <http://laws.deinf.ufma.br/NCLeclipse/pt-br:start/>. Acessado em 22 out. 2014.
- RATAMERO, E.M. Tutorial sobre a linguagem de programação NCL (Nested Context Language). 2007. 11 p. Universidade Federal Fluminense, Niterói, Rio de Janeiro, 2007.
- RODRIGO, R.F.; SOARES, L.F.G. Produção de conteúdo declarativo para TV digital. Campo Grande – MS, Sociedade Brasileira de Computação: XXVI Congresso da SBC, 2006.
- SANTA'ANA, F.; CERQUEIRA, R.; SOARES, L.F.G. NCLUA - Objetos imperativos LUA na linguagem declarativa NCL. Rio de Janeiro – RJ, PUC-Rio: Departamento de Informática, 2008.
- SOARES, L.F.G.; BARBOSA, S.D.J. Programando em NCL 3.0. 1. Ed. Rio de Janeiro: Elsevier, 2009.
- VIRTUALSTB. Ferramentas de Exibição. Disponível em: <http://www.gingaNCL.org.br/ptbr/ferramentas>. Acessado em 22 out. 2014.
- VMWARE. Vmware Player. Disponível em <http://www.vmware.com/br>. Acesso em 22 out. 2014.
- WINSCP. Free SFTP, SCP and FTP client for Windows. Disponível em: <http://winscp.net/eng/index.php>. Acesso em 22 out. 2014.