

## DESENVOLVIMENTO DE UMA APLICAÇÃO PARA TV DIGITAL COM DIVULGAÇÃO DE INFORMAÇÕES SOBRE OS CURSOS DE UMA INSTITUIÇÃO DE ENSINO SUPERIOR

Carlos Marques<sup>1</sup>  
Elyssandro Piffer<sup>1</sup>  
Izabel Cristina Mioranza<sup>1</sup>  
Leandro Clementino de Lima<sup>1</sup>  
Anderson Pereira de Souza<sup>2</sup>  
Camila Coneglian Freitas<sup>2</sup>  
Renan Haertel<sup>2</sup>

MARQUES, C.; PIFFER, E.; MIORANZA, I. C.; LIMA, L. C. de. Desenvolvimento de uma aplicação para tv digital com divulgação de informações sobre os cursos de uma instituição de ensino superior. **Rev. Ciênc. Empres. UNIPAR**, Umuarama, v. 19, n. 1, p. 41-55, jan./jun. 2018.

**RESUMO:** Após a adoção do Sistema Brasileiro de Televisão Digital, apresentou-se a sociedade brasileira uma série de oportunidades para a divulgação de conteúdos e principalmente a disseminação da informação, fazendo que esta possa chegar aos mais diferentes locais, promovendo uma maior inclusão social. Através da interatividade é possível colocar à disposição do usuário da Televisão Digital, informações relevantes que podem facilitar o seu cotidiano, impactando assim, na sua qualidade de vida. Objetivando ampliar o estudo desta nova tecnologia, focando os principais aspectos envolvidos para o desenvolvimento de um aplicativo neste paradigma emergente e oportunizando as comunidades abrangidas na área de atuação da Universidade Paranaense - UNIPAR, foi implementada uma aplicação utilizando linguagem declarativa NCL, aliada a linguagem procedural LUA, que têm seus fundamentos abordados no decorrer do texto. A aplicação visa oportunizar informações sobre os cursos ofertados pela UNIPAR, tanto na graduação como na pós-graduação.

**PALAVRAS-CHAVE:** Televisão, digital, interatividade, cursos, programação, graduação e pós-graduação.

---

DOI: 10.25110/receu.v19i1.6440

<sup>1</sup>Professor do curso de Sistemas de informação da Universidade Paranaense, UNIPAR, unidade de Umuarama.

<sup>2</sup>Acadêmicos do curso de Sistemas de Informação da Universidade Paranaense, UNIPAR, unidade de Umuarama.

## DEVELOPMENT OF A DIGITAL TV APPLICATION WITH THE PROMOTION OF COURSES FROM A HIGHER EDUCATION INSTITUTION

**ABSTRACT:** After the adoption of the Brazilian Digital Television System, the Brazilian society has been presented with a number of opportunities for promoting content and mainly the dissemination of information, making it reach many different places, thus promoting greater social inclusion. Interactivity has made it possible to provide Digital TV users with relevant information to ease their daily lives, with impact on their quality of life. Aiming at expanding the study of this new technology, focusing on the main aspects involved in the development of an app in this emerging paradigm and providing opportunities for communities in the operating area of Universidade Paranaense - UNIPAR, an app was implemented using the NCL declarative language, combined with the procedural language LUA, which will be addressed throughout the paper. The app aims at providing information on the courses offered by Unipar, at both undergraduate and graduate levels.

**KEYWORDS:** Course; Digital; Interactivity; Program; TV; Undergraduate and graduate.

## DESARROLLO DE UN APLICATIVO PARA TV DIGITAL CON DIVULGACIÓN DE INFORMACIONES SOBRE LOS CURSOS DE UNA INSTITUCIÓN DE ENSEÑANZA SUPERIOR

**RESUMEN:** Tras la adopción del Sistema Brasileño de Televisión Digital, se presentó a la sociedad brasileña una serie de oportunidades para divulgación de contenidos y, principalmente la diseminación de informaciones, haciendo con que ésta llegue a los más distintos locales, promoviendo mayor inclusión social. A través de la interactividad es posible poner a la disposición del usuario de la Televisión Digital informaciones relevantes que puedan facilitar su cotidiano, impactando así en su calidad de vida. Buscando ampliar el estudio de esta nueva tecnología, y apuntando los principales aspectos involucrados para el desarrollo de un aplicativo en este paradigma emergente y oportunidad a las comunidades abarcadas en el área de actuación de la Universidade Paranaense – UNIPAR, se ha implementado una aplicación utilizando lenguaje declarativa NCL, aliada al lenguaje procedural LUA, que tiene sus fundamentos abordados a lo largo del texto. La aplicación busca dar oportunidad de informaciones sobre los cursos ofrecidos por UNIPAR, tanto en graduación como en postgrado.

**PALABRAS CLAVE:** Cursos; Digital; Graduación y postgrado; Interactividad; Programación; Televisión.

## 1 INTRODUÇÃO

A adoção do Sistema Brasileiro de Televisão Digital – SBTVD trouxe para a comunidade vários benefícios, dentre estes, o recurso de interatividade que possibilita a disseminação de informações utilizando uma tecnologia de ampla difusão e que é utilizada pela maioria da população brasileira. Neste contexto, com o intuito de auxiliar a população a conhecer melhor as opções de cursos superiores e de pós-graduação *lato e stricto sensu* ofertados por uma IES (Instituição de Ensino Superior) em uma determinada região, foi desenvolvida uma aplicação utilizando o *middleware Ginga*, com o intuito de disponibilizar de forma fácil todo este conteúdo, auxiliando os indivíduos interessados a obterem maiores informações e incentivando-os a prosseguir com suas formações profissionais, tão necessárias atualmente.

O SBTVD já é uma realidade em grande parte do território brasileiro, principalmente em regiões com maior índice demográfico e pode se tornar um grande aliado na disseminação do conhecimento. Através deste sistema, não só a imagem e o som de melhor qualidade são recebidos nos mais diversos locais do país, mas também existe a possibilidade da difusão de informações de interesse da comunidade localizada na área de abrangência dos locais em que já se estabeleceu.

Através do *middleware Ginga*, é possível implementar aplicações que podem ser executadas nos aparelhos de televisão compatíveis com este recurso. Essas aplicações podem ter os mais variados intuítos, seja para lazer, entretenimento e até para informação, como a aplicação desenvolvida e demonstrada neste documento.

A aplicação desenvolvida apresenta dados básicos, porém imprescindíveis para que a população interessada possa se informar sobre os cursos de ensino superior e também de *lato e stricto sensu* ofertados pela IES em suas unidades. Informações sobre as unidades e seus respectivos endereços, telefones e outras formas de contato são disponibilizadas de forma simples e intuitiva. Também são apresentadas informações sobre os cursos ofertados em cada unidade.

Para o desenvolvimento deste projeto, foram necessários alguns recursos técnicos, assim como o próprio *Ginga* e duas linguagens de programação: *Ginga-NCL* e *Lua*. O *Ginga-NCL* é uma linguagem declarativa que tem como principal objetivo o posicionamento dos objetos na tela, bem como o controle do fluxo da aplicação, ficando para a linguagem Lua o trabalho do tratamento dos dados, principalmente no que tange as pesquisas de conteúdo a serem disponibilizadas para o usuário (NCL, 2015). Dessa forma aproveita-se o que existe de melhor em cada uma das linguagens, transformando esta união em uma espécie de “simbiose”, fazendo o projeto apresentar melhores resultados.

Visando uma maior proximidade do projeto com a sua aplicabilidade real, tomou-se por base uma instituição de ensino superior (UNIPAR – Universidade Paranaense) em atividade e que possui unidades em sete cidades dispersas nas regiões noroeste, oeste e sudoeste do estado do Paraná.

## 2 DESENVOLVIMENTO

Durante o processo de desenvolvimento foram utilizadas algumas ferramentas que possibilitaram a produção do aplicativo, facilitando as suas tarefas inerentes, entre estas pode-se destacar:

- *VMWare Player*: trata-se de um software que possibilita a execução de uma máquina virtual que tem a função de simular um *set-top box* para o ambiente declarativo *Ginga-NCL*. Nessa ferramenta é que são realizados os testes.
- *WinSPC*: ferramenta de transferência de arquivos via conexões *SSH (Secure Shell)*. Ela é utilizada para transferir os documentos *NCL* e *Lua* para a máquina virtual que simula o *Ginga-NCL*.
- *IDE Eclipse*: *IDE* escolhida para o desenvolvimento, pois oferece *plug-ins* que possibilitam o desenvolvimento tanto para *NCL* quanto *Lua*.
- *Plugin NCL Eclipse*: *plugin* utilizado na *IDE* para a linguagem *NCL*. (NCL, 2015)
- *Plugin Lua Eclipse*: *plugin* utilizado na *IDE* para a linguagem *Lua*.

Com todas as ferramentas descritas anteriormente devidamente instaladas, o ambiente de desenvolvimento ficou apto para o início de implementações para a televisão digital, bastando apenas realizar algumas configurações para que os arquivos da aplicação pudessem ser enviados para a máquina virtual e posteriormente executados por ela.

O primeiro passo para iniciar o desenvolvimento foi criar um projeto em linguagem *NCL* dentro do *IDE Eclipse*, acessando o caminho “File/New/Project” e selecionar um projeto do tipo *NCL*. Definiu-se o nome para o projeto, o local de seu armazenamento, foram marcadas as opções para criar um diretório para os arquivos de mídia e a adicionada uma base de conectores padrão do *NCL Eclipse*.

Com o projeto *NCL* criado, por meio do *NCL Eclipse* foi inserido um arquivo do tipo *NCL* para conter a aplicação em si. Neste arquivo, pode-se constatar que o *NCL Eclipse* criou uma estrutura básica do arquivo com as *tags* *<ncl>*, *<head>* e *<body>*.

O próximo passo foi criar um arquivo *Lua*, informando em qual projeto

seria criado o arquivo e o seu nome. Neste arquivo foi escrito o código na linguagem *Lua*.

Após a aplicação ser desenvolvida, foi enviada para a máquina virtual para que sua execução pudesse ocorrer. Os arquivos da aplicação foram enviados para a máquina virtual utilizando o programa *WinSCP* com uma conexão SFTP (*Secure File Transfer Protocol*). Dentro do *WinSCP* foi informado o *IP* da máquina virtual no *Host Name*, e por padrão, o usuário “*root*”, senha “*telemidia*”. Uma vez a conexão estabelecida, a pasta da aplicação desenvolvida no *Eclipse* foi enviada para a máquina virtual na pasta “*/misc/ncl30*”.

Com a aplicação devidamente enviada para a máquina virtual, foi necessário realizar uma última configuração dentro do *IDE Eclipse* para que enviasse o comando para executar a aplicação. Dentro do *Eclipse*, acessou-se o caminho “*Window/Preferences*”, na tela que foi aberta, localizado no menu “*NCL*” o sub-menu “*Remote Ginga-NCL Play*”. Nesta tela de configuração, foram alterados os parâmetros “*IP da máquina remota*”, “*Nome do usuário*” e “*Senha*”, para as informações da máquina virtual.

Após esses procedimentos, bastou executar a aplicação diretamente do *Eclipse*, visualizando o resultado na máquina virtual do *Ginga*. Durante o desenvolvimento e os respectivos testes, a cada vez que uma alteração era efetuada, os arquivos foram reenviados para a máquina virtual.

## 2.1 Linguagem NCL

A linguagem *NCL* é semelhante a linguagem de marcação *XML* (*eXtensible Markup Language*), ou seja, é declarativa e trabalha baseada com seus elementos ou *tag's*. O *<ncl>* é o elemento principal e posteriormente os elementos *<head>*, *<body>*, que são filhos do mesmo (SOARES, 2009).

### 2.1.1 Region

O elemento *region* é o responsável por estabelecer o local que cada elemento de mídia (áudio, vídeo, imagem e etc.) será apresentado. Os atributos do elemento *region* são: *id*, *left*, *top*, *right*, *bottom*, *width*, *zIndex*. Todos eles recebem a função de indicar qual será o posicionamento da mídia, aceitando valores em *pixels* ou em porcentagem. A figura 1 demonstra um exemplo de uso do elemento *<region>* que foi utilizado no projeto.

Figura 1: Definição do elemento *region*.

```
<region id="RgTLogo" left="82.00%"
top="20.00%" width="15.00%" height="60.00%"
zIndex="4"/>
```

Fonte: Elaborado pelos autores, (2017).

Na aplicação é utilizada apenas uma *regionbase*, composta de diversas *regions*, especificando a parte da tela onde será apresentada a mídia. Dentre elas destacam-se:

- *RgTopo*: responsável por definir um tamanho máximo para apresentação dos elementos do menu do topo. Esta *region* serve para agrupar para as regiões *RgTBT1*, *RgTBT2*, *RgTBT3* e *RgTLogo*, que são respectivamente a opção 1, 2 e 3 do menu, seguindo do logo da instituição de ensino.
- *RgRodape*: contém a posição do rodapé da aplicação que contempla as informações de controle para auxiliar o usuário navegar dentro da aplicação.
- *RgRDir*: utilizada para demonstrar como é feita a navegação das opções pelo controle remoto.
- *RgROk*: utilizada para demonstrar como é feita a seleção das opções pelo controle remoto.
- *RgRSair*: utilizada para demonstrar como é feito para sair da aplicação pelo controle remoto.
- *RgVideo*: utilizada na apresentação do vídeo que simula a programação normal do canal de mídia.
- *RgFundo*: apresenta uma imagem de fundo quando a aplicação for iniciada.
- *RgInfo*: apresenta as imagens dos cursos, unidades ou níveis pela aplicação.
- *RgCentro*: responsável por definir um tamanho máximo para apresentação dos elementos do menu do meio da aplicação. Esta *region* serve como mãe para as regiões *RgCBTEsq*, *RgCBT1*, *RgCBT2*, *RgCBT3*, *RgCBT4*, *RgCBT5* e *RgCBTDir*, que são respectivamente o botão esquerdo da navegação do menu do meio, as colunas 1, 2, 3, 4, 5 do menu do meio e o botão direito de navegação do menu.
- *RgBcurso*: apresenta as informações selecionadas pelo telespectador, que serão carregadas pela aplicação, como as informações dos cursos, níveis e unidades.

### 2.1.2 Descritor

O *descriptor* é o elemento responsável pela ligação entre um objeto de mídia e uma região. Ele é representado pela *tag* `<descriptor>` seguido do atri-

buto de identificação *id* e a definição da região que está relacionada a esse elemento. Alguns dos atributos do elemento são: *id*, *region*. A figura 2 demonstra o exemplo de utilização deste recurso no projeto.

Figura 2: Definição do elemento *descriptor*

```
<descriptor id="DescTLogo" region="RgTLogo"
focusIndex="11" moveDown="12" moveLeft="3"
moveRight="1" focusBorderColor="white"/>
```

Fonte: Elaborado pelos autores, 2017.

Atributos utilizados e suas respectivas funções:

- *id*: identificador único.
- *region*: nome da região onde o descriptor está relacionado.
- *focusIndex*: representa o índice sequencial da navegação das teclas do controle remoto.
- *moveUp*: caso seja acionada a tecla para cima representa o destino que o foco será direcionado.
- *moveDown*: caso seja acionada a tecla para baixo representa o destino que o foco será direcionado.
- *moveLeft*: caso seja acionada a tecla para esquerda representa o destino que o foco será direcionado.
- *moveRight*: caso seja acionada a tecla para direita representa o destino que o foco será direcionado.
- *focusBorderColor*: cor da borda do elemento.

### 2.1.3 Conector Base

Para que a aplicação em *NCL* possa executar determinadas ações, é necessário que seja feita a definição de um *connectorbase*, pois ele possui uma base de conectores cadastrados, que serão utilizados na aplicação. Um *connector* define suas ações por meio de seus papéis (*roles*). O *NCL* disponibiliza por padrão um grande acervo de conectores, a partir do qual foi importado utilizando a função *importbase* e utilizado na aplicação conforme apresentado na Figura 3.

Definições das *tags*:

- *<connectorbase>*: a base de conectores, é utilizada para agrupar e ancorar todos os conectores presentes na aplicação.
- *<importbase>*: tag utilizada para a importação do arquivo de conectores do *NCL*, neste caso o arquivo utilizado é o *ConnectorBase.ncl*, presente na mesma pasta do arquivo fonte e está recebendo o apelido de *conn2*.

Figura 3: Definição do *connectorBase*

```
<connectorBase id="connBase">
  <importBase
    alias="conn2"
    documentURI="ConnectorBase.ncl"/>
</connectorBase>
```

Fonte: Elaborado pelos autores, 2017.

### 2.1.4 Mídia

A Mídia é um elemento muito importante na aplicação e é responsável pela definição dos objetos de mídia (áudio, vídeo, imagem, *NCLua* e etc.) que o projeto irá conter. Ela é representada pela *tag* `<media>` seguida do atributo de identificação *id*, o caminho onde a mídia está localizada no projeto e o `<descriptor>` que está relacionado a esse elemento. Alguns dos atributos do elemento são: *id*, *scr*, *descriptor*. Foram utilizados no projeto vários elementos de mídia, principalmente imagens, e objetos *NCLUA*, em que foi adotado um padrão para definição das mídias, sendo que é apresentado um breve exemplo na figura 4.

Figura 4: Definição do elemento *media*.

```
<media id="MCBTesq"
  src="media/img/setaesquerda.png"
  descriptor="DescCBTesq"/>
<media id="MCBTesqL"
  src="media/lua/BtEsq.lua"
  descriptor="DescCBTesq"/>
```

Fonte: Elaborado pelos autores, 2017.

- *id*: Identificador único.
- *scr*: caminho que especifica o elemento utilizado.
- *descriptor*: definição do elemento descritor associado.

### 2.1.5 Propriedade

O elemento responsável por definir as propriedades gerais da mídia é representado pela *tag* `<property>`, esse elemento é filho da *tag* `<media>`, seguido do atributo de especificação *name*. Na aplicação, a propriedade é utilizada para acessar alguns métodos específicos dos vários tipos de objetos de mídia, como por exemplo, na mídia *vídeo*, a *property* recebe nome de *bounds*. Isto sinaliza ao *NCL* que futuramente será alterada a propriedade ligada ao tamanho desta mídia. Cada tipo de mídia tem suas próprias propriedades, sendo necessário observar essas informações no momento de sua definição.

Figura 5: Definição do elemento *property* na mídia *video*

```
<media id="video"
  src="media/videos/video1.mp4"
  descriptor="DesVideo">
  <property name="bounds"/>
</media>
```

Fonte: Elaborado pelos autores, 2017.

### 2.1.6 Area

A *area* é o elemento filho de *<media>* que é responsável por definir um apelido ao elemento. Os atributos definidos no elemento *<area>* são o atributo de identificação *<id>* e seu apelido *<label>*. Essas informações serão futuramente utilizadas nos códigos em *NCLua* para realizar a identificação de qual objeto chamou o código *NCLua*. Por exemplo, a mídia *mTBT1* que é apresentada na primeira posição do menu do topo da aplicação tem seu *label* definido como *Topo1*, onde esta informação será utilizada nos códigos *NCLua* para definir como a aplicação irá se comportar com a chamada do arquivo por este elemento.

Figura 6: Definição do elemento *área*

```
<media id="mTBT1"
  src="media/img/TBT.png"
  descriptor="DescTBT1">
  <area id="TBT1"
    label="Topo1"/>
</media>
```

Fonte: Elaborado pelos autores, 2017.

### 2.1.7 Porta do contexto

A porta é o elemento responsável por indicar qual será a mídia exibida no início da transmissão. Ela é representada pela tag *<port>* seguida do atributo de definição *id*. Os atributos do elemento são: *id* e *component*. A porta utilizada na aplicação, definida com *id* *p0*, define que seu componente *video* será apresentado assim que a aplicação for iniciada, conforme Figura 7.

Figura 7: Definição do elemento *port*

```
<port id="p0" component="video"/>
```

Fonte: Elaborado pelos autores, 2017.

### 2.1.8 Elo

O elo é identificado pela tag `<link>`, sendo o atributo responsável por executar as ações definidas nos conectores, em que são passadas as mídias e suas determinadas tarefas (*roles*). O seu comportamento é definido pelo elemento conector em que está atrelado. O *NCL* define os atributos *id* e *xConnector* como atributos do elemento `<link>`. Sendo assim, é responsável pelo agrupamento e ancoragem das definições das interatividades que serão realizadas a partir de uma ação do usuário, momento de vídeo ou outro tipo de evento. O *NCLua* também interage com o documento através dos elos, que são assim definidos:

- *id*: identificador único.
- *xconnector*: informa a URL que define a relação que é referenciada pela base de conectores `<connectorBase>` definida automaticamente na aplicação e utilizada no elemento `<link>`.
- `<bind>`: elemento complementar que define o que acontece com a interatividade dentro do elemento link, para que todas as funções sejam executadas.
- `<bindParam>`: define as características da execução do *link*.

A figura 8 demonstra parcialmente como exemplo do uso de links, o link *lMenu* que tem como seu *xconnector* o valor `conn2#onKeySelectionSet_varStart` onde define que ao pressionar um botão, no caso da aplicação o botão *MENU*, serão iniciadas diversas mídias, como os botões da aplicação e o fundo, e será definido que no componente *vídeo*, em sua interface *bounds* serão atribuídos os valores `5%,2%,20%,20%`, isso ocasionará no redimensionamento do vídeo principal, para que não ocupe todo o espaço da aplicação, abrindo espaço para a apresentação dos botões e outros elementos.

Figura 8: Definição do elemento *link lMenu*.

```
<link id="lMenu"
  xconnector="conn2#onKeySelectionSet_varStart">
  <bind role="onSelection" component="video">
    <bindParam name="keyCode" value="MENU"/>
  </bind>
  <bind role="start" component="curso"/>
  ...
  <bind role="set" component="video"
    interface="bounds">
    <bindParam name="var" value="5%,2%,20%,20%"/>
  </bind>
</link>
```

Fonte: Elaborado pelos autores, 2017.

## 2.2 Lua

A linguagem de *script Lua* desde o seu desenvolvimento nos anos 90 tem como objetivo ser utilizada em conjunto com outras linguagens, funcionando como um satélite, que auxilia e complementa o desenvolvimento (LUA, 2016). Justamente por esta característica, dificilmente é encontrado um projeto desenvolvido inteiramente em *Lua*, portanto foi escolhida como linguagem complementar ao ambiente declarativo do *Ginga-NCL*. Essa parceria foi denominada de *NCLua*. Suas principais características são: ser *open source*, simples, robusta, rápida, portátil e embarcada.

### 2.2.1 Arquivos Topo

A aplicação contempla vários arquivos *Lua*, com diversas finalidades, porém o de maior importância é o *BoxImp.lua* que é responsável por trazer para a tela as informações solicitadas pelo usuário e obtidas dos arquivos de dados.

Inicialmente verifica-se qual dos botões foi pressionado de forma a possibilitar a seleção correta do texto a ser impresso e a variável *opcao* recebe o valor referente ao botão pressionado. Em seguida é realizada a abertura do arquivo de caminho presente na variável *arquivo* para a busca da *string* de busca dentro dele.

Figura 9: *BoxImp.lua*

```
function LuaNCL (evt)
  if evt.label == 'Centro1' then opcao = 1
  elseif evt.label == 'Centro2' then opcao = 2
  elseif evt.label == 'Centro3' then opcao = 3
  elseif evt.label == 'Centro4' then opcao = 4
  else opcao = 5
  end
  spos = btdir * 5 + opcao
  cursoarq = io.open(Arquivo, 'r')
  cursoarq:seek('set')
  cursopos = spos .. ';'
  linha = 0
  for linha in io.lines(cursoArquivo) do
    cursotexto = cursoarq:read('*line')
    pos = string.find(cursotexto, cursopos, 1)
    if (pos) then
      pos = string.find(cursotexto, ';', 1)
      pos2 = string.find(cursotexto, ';', pos+1)
      pos3 = string.find(cursotexto, ';', pos2+1)
      texto = string.sub(cursotexto, pos2+1,
pos3-1)
      break
    end
  end
  nomeArquivo =
"/misc/ncl30/PIC2015/media/txt/Imprimir.txt"
  info = assert(io.open(nomeArquivo, "w+"),
"Arquivo não pode ser criado")
  info:seek("set")
  info:write(texto)
  info:close()
  event.post {
    class = 'ncl',
    type = 'presentation',
    action = 'stop'
  }
end
event.register(LuaNCL)
```

Fonte: Elaborado pelos autores, 2017.

### 2.3 Funcionamento da Aplicação Desenvolvida

A premissa principal no desenvolvimento dessa aplicação é de deixá-la o mais simples possível, minimizando os passos necessários para o usuário chegar até as informações desejadas. Dessa forma, um número maior de pessoas pode se beneficiar das informações disponibilizadas.

Uma vez que a aplicação esteja disponível para acesso, o aparelho receptor demonstrará isto por meio de uma imagem que aparecerá no canto esquerdo da tela, juntamente com o programa televisivo que estiver sendo transmitido naquele momento. Neste momento o usuário deverá pressionar o botão correspondente no controle do aparelho receptor para acessar a aplicação.

Quando a aplicação começar a ser executada, a imagem será redimensionada e o programa televisivo que estiver sendo veiculado continuará a ser mostrado para o usuário, porém em uma tela de menor dimensão, conforme figura 10.

Figura 10: Aplicação iniciada



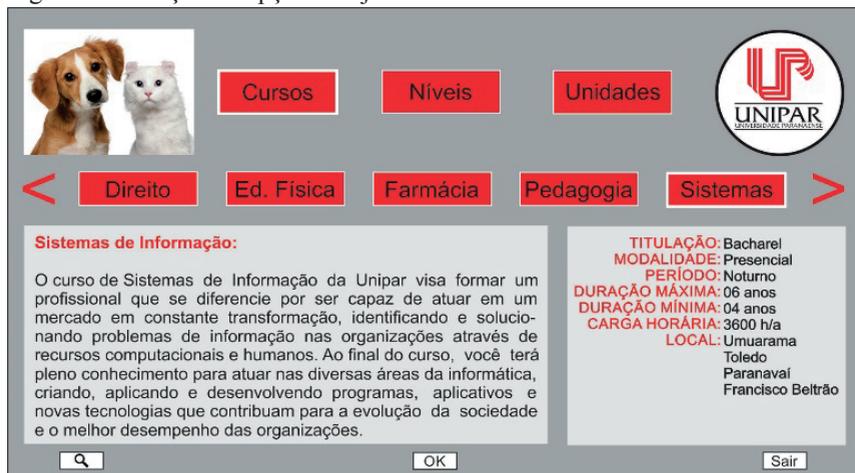
Fonte: Elaborado pelos autores, 2017.

Neste momento, conforme a figura 10 serão disponibilizadas para o usuário, três opções: Cursos, Níveis e Unidades. Estas opções estarão em forma de botões onde o usuário deverá selecionar a opção desejada por meio 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 11.

Figura 11: Seleção da opção desejada - Cursos



Fonte: Elaborado pelos autores, 2017.

Ao selecionar os cursos, será mostrada uma barra com todos os cursos dispostos na horizontal, apresentando duas setas direcionais (uma à esquerda e outra à direita) possibilitando a navegação entre todos os cursos, uma vez que não é possível mostrar todos eles em uma única tela.

A figura 11 demonstra a tela a ser disponibilizada ao usuário quando este seleciona um curso, disponibilizando uma série de informações. Dependendo do item selecionado, os dados apresentados serão diferentes, pois informações sobre cursos, níveis e unidades, obviamente, apresentam características próprias.

Assim como para os cursos, as opções para os níveis (níveis de formação ofertados pela instituição) e também para as unidades da instituição, seguem o mesmo padrão, tanto para a seleção dos itens desejados como de tela de apresentação das respectivas informações.

As informações disponibilizadas pela aplicação não são vinculadas, sendo transmitidas em arquivos separados. Com a adoção de arquivos separados da aplicação, a instituição poderá a qualquer momento, efetuar as devidas atualizações nos arquivos e os enviar para a emissora de TV que substituirá os arquivos antigos pelos novos, sem a necessidade de intervenção de um programador, tornando assim um sistema de fácil manutenção.

### 3 CONSIDERAÇÕES FINAIS

O potencial que a interatividade apresenta para um país como o Brasil, com dimensões continentais pôde ser evidenciado durante o desenvolvimento deste trabalho, principalmente quanto à transmissão de conhecimento e informação.

Obviamente muitas pessoas poderão se beneficiar deste fato mediante a popularização da TV Digital, bem como a crescente migração do sistema de transmissão analógica para esta nova tecnologia.

Porém, ainda há muito a ser feito, pois 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.

Outro fator a ser discutido é a tecnologia envolvida, em que se pode perceber que apesar de algo novo, o uso das linguagens NCL e Lua é relativamente fácil e pode render bons resultados finais.

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 para inserir a aplicação e os dados, juntamente com o conteúdo que já é transmitido (vídeo e áudio), não foi possível até o momento. Ainda há o contato com a operadora, a qual disse ter interesse em viabilizar o projeto por meio da aquisição destes equipamentos num futuro próximo.

É visível a necessidade de aprimoramento acerca dos estudos sobre o contexto de TV Digital e sobre o desenvolvimento de aplicações interativas com a utilização da linguagem declarativa *NCL*. Existem pontos positivos e negativos, mas o principal é a função que a aplicação desenvolvida exercerá sobre a comunidade, possibilitando uma maior visibilidade dos cursos ofertados, pela instituição de ensino, ajudando a todos na escolha de seu futuro acadêmico e profissional.

Outro aspecto de suma importância foi a colaboração e a troca de experiências que o projeto proporcionou junto à comunidade de desenvolvimento *NCL*, compartilhando as experiências, conteúdos técnicos e científicos no decorrer do mesmo, proporcionado assim uma maior disseminação do conhecimento envolvido.

### REFERÊNCIAS

**LUA. A Linguagem de Programação LUA.** URL: <http://www.lua.org/portugues.html>. Acesso em: 21/03/2016.

NCL. **Visão geral**. URL: <http://www.ncl.org.br/pt-br/inicio>. Acesso em: 03/02/2016.

SOARES, L. F. G.; BARBOSA, S. D. J. In: **Programando em NCL 3.0: Desenvolvimento de Aplicações para o Middleware Ginga, TV Digital e Web**. Rio de Janeiro: Elsevier, 2009.