

Desenvolvendo para TV Digital

Alexandre Belchior Medeiros

Associação Educacional Dom Bosco
alexbelchior@gmail.com

Fabiano Cavassani Notz

Associação Educacional Dom Bosco
fabiano.cavassani@yahoo.com.br

Gilber Leite Leal

Associação Educacional Dom Bosco
gilberleal@hotmail.com

Gabriel P. S. Brenner

Associação Educacional Dom Bosco
gabrielbrenner@gmail.com

RESUMO

O artigo abordará o desenvolvimento de aplicações voltadas para a TV Digital Interativa e as tecnologias, ambiente de desenvolvimento as principais linguagens utilizadas, como instalar as ferramentas de desenvolvimento, apresentando ainda um tutorial de como fazer uma aplicação para TV Digital.

Ao final deste artigo, o leitor terá as informações básicas e necessárias de como começar neste novo segmento que cresce a cada dia e que tem um potencial gigantesco de um futuro mercado de trabalho, que demandará de mão de obra qualificada nas tecnologias que serão apresentadas.

Palavras-Chave: TV Digital Interativa; NCL; Lua.

1. INTRODUÇÃO

O sistema brasileiro de TV Digital Terrestre, formado pela junção da tecnologia japonesa, o ISDB-T (Serviço Integrado de Transmissão Digital Terrestre) com os estudos realizados pelas instituições de nível superior e empresas de interesse na tecnologia, possui uma especificação aberta. Diante deste fato e da expansão do sinal de TV Digital aberta no Brasil, surgem às primeiras aplicações direcionadas à TV Digital Interativa. Dentre os principais desenvolvedores destacam-se a Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) e o Laboratório de Aplicações de Vídeo Digital da Universidade Federal da Paraíba (LAViD -UFPB) nas quais contribuíram consideravelmente para o desenvolvimento do Ginga, um *middleware* que juntamente com as linguagens de programação NCL e Lua, formaram o padrão do Sistema Brasileiro de TV Digital Interativa conforme a norma ABNT NBR 15606-1:2010.

Como toda tecnologia nova, os profissionais que desejam se aprofundar nessa área se deparam com um universo de normas e informações (API's, plataformas, práticas de programação, ferramentas, etc) nas quais podem trazer transtornos e grandes dificuldades com a escassez de material explicativo e ambientes ainda não completamente maturados para desenvolvimento e testes.

Alguns princípios para o desenvolvimento de aplicações para TVDI estão melhores detalhados na seção 2 onde será apresentado o ambiente de desenvolvimento para TVDI; a seção 3 trará detalhado, um passo a passo da instalação das ferramentas utilizadas, para que o ambiente de desenvolvimento e testes possa ser montado; a seção 4 apresentará as linguagens de programação utilizadas pelo padrão brasileiro, sendo aplicadas na seção 5, onde um exemplo de aplicação para TV Digital será construído.

2. AMBIENTE PARA DESENVOLVIMENTO DE APLICAÇÕES DE TV DIGITAL

Nesta seção serão apresentadas as ferramentas utilizadas atualmente no desenvolvimento de aplicações para TV Digital.

Todas as ferramentas exploradas neste artigo são gratuitas e de código aberto, o que proporciona uma evolução constante.

2.1 ECLIPSE

O eclipse é uma IDE (*integrated development environment*) gratuita, de um projeto *open source* no qual ganhou força através da multinacional IBM que o utiliza como base no desenvolvimento de algumas de suas ferramentas comerciais. Segundo as comunidades de desenvolvedores Java o eclipse está no topo da lista entre as IDE's utilizadas para programação.

Uma ferramenta aberta que é capaz de utilizar inúmeros plugins associados, possui assistente de códigos, PDE: Ambiente de desenvolvimento de plugins, refatoração entre diversas outras qualidades. No caso do desenvolvimento para TV Digital, é utilizado para criação dos documentos NCL utilizando um plugin criado por pesquisadores do laboratório TeleMídia da PUC-Rio, e os arquivos Lua, utilizando o plugin LuaEclipse.

Para a instalação do plugin NCL é aconselhável a utilização do eclipse versão 3.5 ou superior.

2.2 PLUGIN NCLECLIPSE

A mais recente versão do plugin de código livre e voltado ao desenvolvimento de aplicações interativas em NCL utilizado na IDE eclipse, o NCLEclipse 1.5 foi lançado em 23 de março com principal intuito de facilitar a programação de conteúdos interativos para TVDI, trazendo produtividade, facilidade e minimizando os erros.

2.3 PLUGIN LUECLIPSE

É formado por um conjunto de plugins utilizados na plataforma do eclipse. A primeira versão foi criada nos meados de 2003 por uma empresa chamada Ideais Tecnologias e hoje é mantida pela FINEP, SEBRAE e a empresa Fábrica Digital.

Atualmente somente a versão 1.3 do plugin LuaEclipse está homologada para uso, e a versão 2.0 está em desenvolvimento por Edgard Arakaki e Luís Eduardo Jason Santos, com contribuições de André Carregal e Humberto dos Anjos. Este plugin ainda possui menos recursos que o NCLEclipse, porém é ideal para desenvolvimento do NCLua, o *script* criado em Lua para o NCL.

2.4 VMWARE PLAYER 3.1

O *VMware player* é uma ferramenta utilizada para virtualização, que para o desenvolvimento de aplicações de TVDI irá simular a máquina virtual fedora-fc7-ginga-i386, também chamada de *Set-top Box* Virtual Ginga-NCL (STB Ginga-NCL), ou simplesmente o conversor digital com Ginga-NCL embarcado.

2.5 SET-TOP BOX VIRTUAL GINGA-NCL

É uma máquina virtual baseada no sistema operacional Linux, distribuição Fedora Core 7 que foi desenvolvida e configurada pelos pesquisadores do laboratório TeleMídia da PUC-Rio, portanto não é necessário realizar a configuração do *kernel* ou *boot*. Possui uma instalação simples e rápida, sem complicação, mesmo para usuários iniciantes. Sua principal finalidade é emular um ambiente próximo ao *Set-top Box* real. Por utilizar o *hardware* do PC,

o STB Ginga-NCL possui algumas desvantagens, como por exemplo, a resolução de 640x480 pré-configurada para garantir um melhor desempenho na utilização da CPU.

2.6 SSH - SECURE SHELL CLIENT

O SSH é um *software* e ao mesmo tempo um protocolo de rede no qual permite a conexão com outro computador na rede. No caso do ambiente para desenvolvimento de TVDI, ele irá fazer conexão com a máquina virtual que emula o STB Ginga-NCL. O SSH possui também um componente que fará a transferência dos arquivos Lua e NCL armazenados no *workspace* do eclipse.

3. INSTALAÇÃO DAS FERRAMENTAS

3.1 INSTALAÇÃO DO ECLIPSE

Link para download: <http://www.eclipse.org/downloads/>.

Na verdade o eclipse não é instalado e seu pacote é formado por vários arquivos compactados. Descompactando-o em um diretório (mais usual > C:\arquivos de programas) será criado uma pasta chamada eclipse na qual contem diversos arquivos e subpastas. O executável que inicia a IDE é o eclipse.exe, que para facilitar, pode ser acessado através de um atalho da área de trabalho.

Na inicialização do eclipse é exibida uma tela que solicita um diretório para o *workspace* (diretório onde serão criados todos os projetos feitos no eclipse). Por padrão o *workspace* é criado dentro da própria pasta do eclipse. Com o *workspace* definido o eclipse estará pronto para uso.

3.2 INSTALAÇÃO DO PLUGIN NCLECLIPSE

A instalação do plugin NCLEclipse pode ser realizada de duas maneiras, manualmente ou via *update site*. Neste artigo será abordada a instalação via *update site*.

Utilizando a versão do eclipse compatível com o plugin, o eclipse 3.5, acessar a aba Menu > Help> Install New Software na qual irá exibir a seguinte janela:

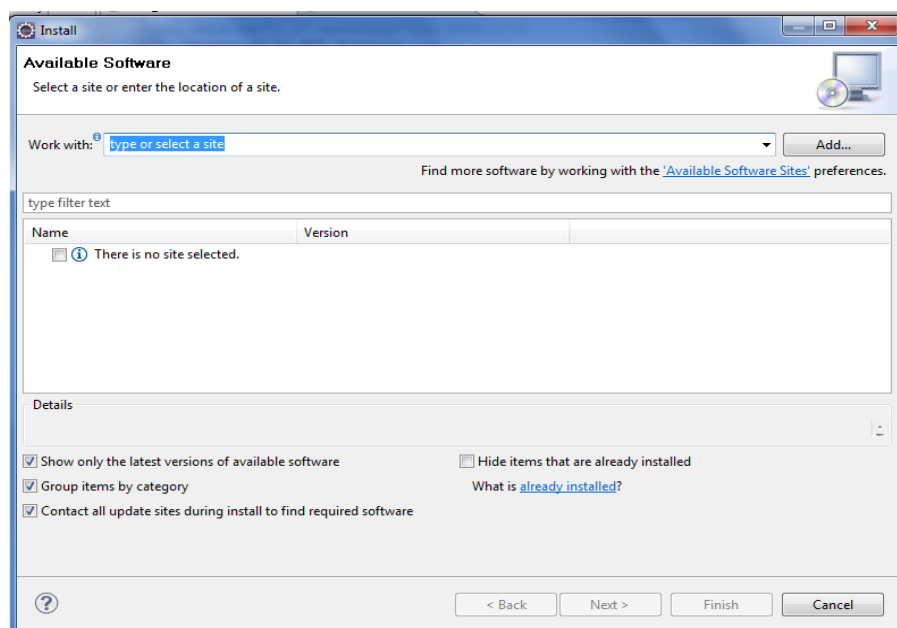


Figura 1: Janela de instalação do NCLEclipse.

Acessando o botão “add” será exibida uma nova janela na qual deverá ser inserido a URL que será acessada para realizar a instalação, <http://www.laws.deinf.ufma.br/ncleclipse/update>. Desmarque a opção “Group Items by category” e clique em “next” até que apareça a licença, aceitando-a o plugin estará pronto para uso.

3.3 INSTALAÇÃO DO PLUGIN LUAECLIPSE

Link para download: <http://luaclipse.luaforge.net/br/index.html#download>.

Um pré requisito para a utilização do LuaEclipse é a instalação da linguagem Lua. A instalação da mesma é simples e descomplicada, basta apenas clicar em “next”. O link para download da linguagem é: <http://www.lua.org/download.html>. Para testar se realmente a linguagem foi instalada com êxito, basta digitar Lua no *prompt* do DOS e irá exibir a versão da linguagem instalada.

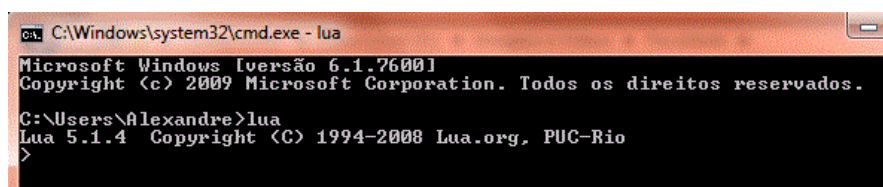


Figura 2: Confirmação de instalação da linguagem Lua.

Assim como o NCLEclipse, o LuaEclipse também possui duas formas de ser instalado. A instalação manual inicia-se com a extração dos arquivos compactados (baixados do link acima) dentro do diretório onde o eclipse foi instalado. A partir daí basta criar um novo projeto no eclipse utilizando a Lua.

Além da instalação do plugin, é necessário configurar um interpretador para linguagem Lua. Na aba *Window*, no item *Preferences > Lua > Installed Interpreters*, mostrará na caixa que fica na lateral direita o local para ser adicionado o interpretador Lua. Clicando em “add” será exibida uma janela onde deverá ser inserido um nome qualquer para o interpretador e abaixo selecionar o local onde está o arquivo executável chamado *lua.exe*, no caso dentro da pasta onde a linguagem Lua foi instalada (normalmente *C:\Arquivos de Programas\Lua\5.1*).

3.4 INSTALAÇÃO DO VMWARE PLAYER

Link para download: <http://www.vmware.com/download>.

Para iniciar a instalação execute o arquivo e clique em “next”, será exibida a janela para selecionar o diretório a ser instalado, logo após poderá ser selecionado os atalhos que serão criados. A instalação encerrará em poucos minutos e o *VMware Player* estará pronto para uso. É recomendável a inicialização da máquina antes do uso.

3.5 INSTALAÇÃO DO SET-TOP BOX VIRTUAL GINGA-NCL

Link para download: <http://www.ncl.org.br/ferramentas/fedora-fc7-ginga-i386.zip>.

O arquivo do download é compactado e deverá ser extraído para um diretório antes de iniciar a instalação.

Abrindo o *VMware Player*, deve-se clicar em “Open a Virtual Machine” para selecionar a máquina virtual que irá simular o *Set-top Box GINGA-NCL*, como descrito anteriormente na seção 2.4. O arquivo a ser selecionado possui o nome de *fedora-fc7-ginga-i386.vmx* e está dentro da pasta que foi extraída do arquivo compactado. Não é necessário

realizar quaisquer configurações, pois a máquina virtual já é pré-configurada de acordo com as necessidades que possam garantir um bom desempenho do *hardware*.

3.6 INSTALAÇÃO DO SSH - SECURE SHELL CLIENT

Link para download: <http://www.vmware.com/Download>.

O SSH está na versão 3.2.9 e sua instalação é tão simples quanto as demais ferramentas constituintes do ambiente para desenvolvimento de aplicações para TV Digital.

Ao executar o arquivo imediatamente inicializará a instalação do SSH, clicando em “*next*” irá exibir a licença da ferramenta, ao aceitar, basta escolher o diretório de instalação, os atalhos a serem criados e em poucos minutos a instalação estará efetuada. No *desktop* será criado dois atalhos, um chamado SSH *Secure Shell Client* no qual executa o SSH que irá conectar ao outro computador, no caso o a máquina virtual onde se encontra o STB (*Set-top Box*). O outro atalho chamado SSH *Secure File Transfer Client* será o componente para transferência dos arquivos do computador para a máquina virtual.

4. REQUISITOS DE HARDWARE

Para a utilização das ferramentas citadas neste artigo é necessário um *hardware* de porte significativo devido ao alto nível de virtualização, decodificações e renderizações de mídias utilizadas, visto que o *hardware* do *Set-top Box* é bem superior e específico para tal finalidade, diferente do Virtual STB Ginga-NCL emulado no computador. A recomendação mínima é: arquitetura Intel com processador Pentium 4 3.0GHz ou superior, memória RAM de 1 Gb (2 Gb é ideal), placa de vídeo com 64Mb de aceleração, *chipsets* nVidia e ATI recomendados, disco rígido com 5Gb livres.

5. LINGUAGENS UTILIZADAS

Após a definição do padrão do Sistema Brasileiro de TV Digital, começaram os esforços em cima das tecnologias que seriam empregadas para o desenvolvimento, as linguagens NCL (Linguagem de Contexto Aninhado), Lua e Java.

O Brasil precisava de uma plataforma que suportasse as linguagens supracitadas, então foi desenvolvido pelos laboratórios de telemídia da PUC-Rio e LAViD -UFPA a plataforma Ginga-NCL e Ginga-J respectivamente, responsável pela interpretação das linguagens e apresentação multimídia na TV.

5.1 LINGUAGEM JAVA

O Java foi escolhido principalmente para o desenvolvimento de aplicações procedurais, diferente de NCL e Lua.

5.2 LINGUAGEM NCL

A NCL é uma linguagem declarativa baseada em XML (Linguagem de Marcação Extensível), é através dela que são definidos os comportamentos, posição dos vários tipos de mídia, ou seja, toda a apresentação em si, a seguir a listagem de tipos de mídia que o Ginga-NCL suporta de acordo com Soares e Barbosa (2009).

Tabela 1: Tipos de mídia suportados pelo Ginga-NCL.

Tipo de mídia	Extensão do arquivo
text/html	htm, html
text/plain	txt
text/css	css
text/xml	xml
image/bmp	bmp

image/png	png
image/jpeg	jpg, jpeg
audio/basic	wav
audio/mp3	mp3
audio/mp2	mp2
audio/mpeg	mpeg, mpg
audio/mpeg4	mp4, mpg4
vídeo/mpeg	mpeg, mpg
application/x-ginga-NCLua	lua
application/x-ginga-NCLet	class.jar
application/x-ginga-settings	no src (source)
application/x-ginga-time	no src (source)

O NCL e o Ginga-NCL já foram aprovados como padrão internacional pela União Internacional de Telecomunicações (UIT) através da recomendação H.761 para atender apresentações multimídias e interatividade na TV.

5.3 LINGUAGEM LUA

A linguagem Lua foi escolhida pelo fato de ter como principal característica poder trabalhar em sistemas embarcados e de ser uma linguagem leve e rápida, ela trabalha junto como NCL em forma de script e é conhecida também como NCLua, além de ter sido desenvolvida no próprio país pela PUC – Rio.

Na norma ABNT NBR 15606-2:2007 tem todos os detalhes para a codificação utilizando as linguagens citadas.

O desenvolvimento para aplicações para TV Digital foge um pouco do desenvolvimento de aplicações convencionais, e é óbvio que tem de ser, por vários fatores, como por exemplo, a exibição da aplicação não é no monitor do computador e sim na TV, não existe *mouse*, existem vários tamanhos de televisão, a aplicação deve ter no máximo 6Mb de tamanho total, porque é o máximo que um *Set-top Box* suporta, entre muitos outros fatores, seja *hardware* ou ainda pela própria limitação das linguagens. Por essas e outras razões o desenvolvimento da aplicação tem de ser simples, bem intuitivo, onde o usuário ou telespectador não encontre dificuldades para o uso da aplicação.

6. EXEMPLO DE APLICAÇÃO PARA TV DIGITAL

Nesta seção será demonstrada a criação de uma aplicação básica que exibe a mensagem “*Hello Word*”.

6.1 PREPARAÇÃO

Ao iniciar o eclipse, foi definida a pasta *workspace Launcher* em uma pasta de fácil localização. Para esse exemplo a pasta “PrimeiroNCLua” foi criada no *desktop* para facilitar o restante das atividades. Após iniciado o Eclipse foi selecionado o item *File* na barra de tarefas.

Na aba que se abre, seguiu-se a seleção dos itens “*New*” e em seguida o item “*Project*” para a criação de um novo projeto.

Na janela “*New Project*” foi selecionado a pasta “*General*” e em seguida o ícone “*Project*”. Após isso foi selecionado o botão “*Next*”.

Após ser dado o nome “ProjetoNCLua” no “*Project name*”, então foi selecionado o botão “*Finish*”.

Com o novo projeto criado, foi iniciada a criação do programa NCL especificamente, com a seguinte sequência:

Selecionado na barra do Eclipse o item “File”. Na aba que se abriu selecionou-se o item “New” e em seguida o “Other...”.

Em seguida, com a janela “New” que surgiu, selecionou-se a pasta “NCL”, e em seguida o ícone “NCL Document”. Para seguir foi clicado no botão “Next”.

Teremos então na janela “NCL Document Wizard” a escolha da pasta Container do arquivo, e para isso foi selecionado o botão “Browse...”

Se pode observar que na janela “Folder Selection” o eclipse já exibiu a pasta criada anteriormente, e o próximo passo foi então selecionar a pasta do projeto. Neste caso a pasta “ProjetoNCLua”, e então selecionou-se o botão “OK”.

Próximo passo foi dar um nome para aplicação, e o nome utilizado foi o “NossoPrograma”. Para isso foi escrito o nome no campo “Id:”. Foi observada a geração automática do “filename:”. Depois disso foi encerrada a criação utilizando o botão “Finish”.

Obs: Não pode esquecer-se de fechar a tela de boas vindas (welcome) do Eclipse.

O resultado do programa criado está na Figura 1.

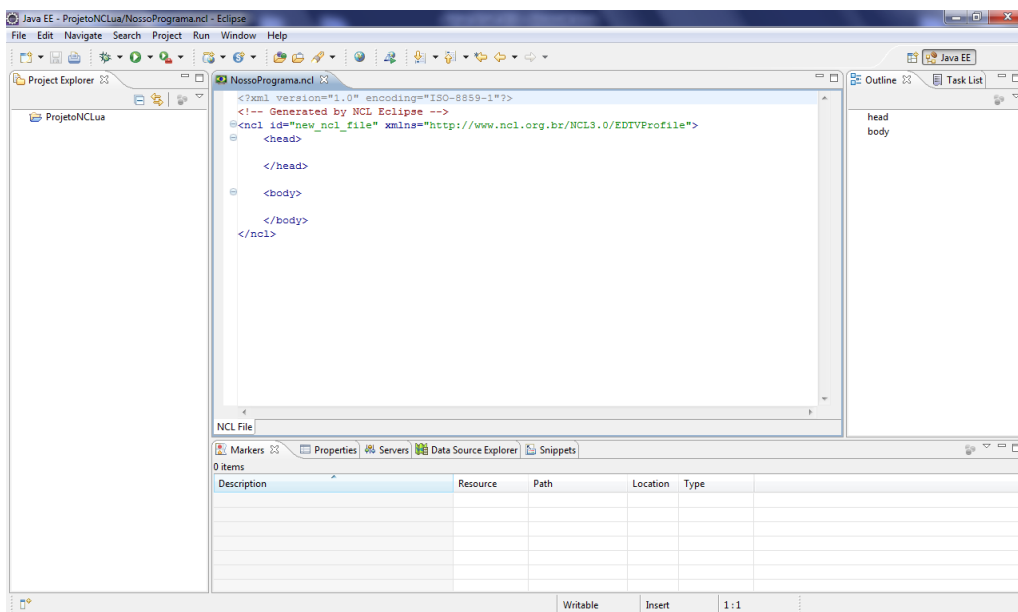


Figura 3: Janela de codificação do Eclipse.

O trecho do programa a seguir já foi gerado automaticamente pelo eclipse.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<nclid="NossoPrograma"
xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
  </head>
  <body>
  </body>
</ncl>
```

E é nesse corpo que se desenvolve a criação do NCL.

6.2 PROGRAMAÇÃO EM NCL

As tags `<head>` `</head>` marcam o trecho que declaramos a região base.

O NCL é uma linguagem que trabalha com tags, que são identificadas pelos sinais “<” e “>” na sua abertura e “</>” no seu fechamento.

A região base é o espaço de tela que o programa utilizará para se exibir. É uma demarcação de área utilizável pela mídia. Entre as tags `<head>` e `</head>` foi inserida a tag `<regionBase>`, e dentro dela foi declarado uma tag de nome `<region>`, que receberá os parâmetros que a região terá segundo o modelo que se segue:

```
<regionBase>
    <region id="rgMain" width="50%" height="50%"
left="30" top="30" />
</regionBase>
```

Os parâmetros que foram passados para a região são os seguintes:

“Id” = Nome identificador da região.

“width” = largura da região.

“height” = altura da região.

“left” = distância da lateral esquerda da tela.

“top” = distância do topo da tela.

Obs.: Com exceção do “Id”, todos os outros parâmetros podem receber porcentagem ou valor em pontos.

Não se pode esquecer-se de fechar a tag `<regionBase>`, reescrevendo o seu nome e acrescentando a barra “/” antes do nome. Desta forma:

```
</regionBase>
```

Será descrito agora as Bases do programa.

Iniciando a tag `<descriptorBase>` foi associado à região de nome “rgMain” que foi associado ao descritor de id=”dsMain” para poder associá-lo com a “mídia” futuramente, conforme o exemplo.

```
<descriptorBase>
    <descriptor id="dsMain" region="rgMain"/>
</descriptorBase>
```

Lembrando que em um programa NCL, é chamado “Mídia” todos os tipos de arquivos, como vídeos.mpg, fotos.jpg e programas.lua.

A próxima etapa é a declaração da porta, que é por onde o programa inicia a sua aplicação.

Na tag `<body>` foi aberta uma tag `<port>` para associar um “id” de uma mídia como componente da seguinte forma:

```
<port id="portainicio" component="luaMain"/>
```

O “portainicio” foi o nome dado ao “id” que iniciará a aplicação usando o componente “luaMain”.

Na criação desta porta o eclipse emite um ícone de erro devido à criação do arquivo lua ainda não ter sido executada.

Por último foi associar a “media” a um arquivo Lua que foi nomeado de main.lua e também foi associado a “media” o descritor que foi declarado como “dsMain”. Segue o exemplo:

```
<media id="luaMain" src="main.lua" descriptor="dsMain" />
```

O programa completo ficou da seguinte forma:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="NossoPrograma"
xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <regionBase>
      <region id="rgMain" width="50%" height="50%"
left="30" top="30" />
    </regionBase>
    <descriptorBase>
      <descriptor id="dsMain" region="rgMain"/>
    </descriptorBase>
  </head>
  <body>
    <port id=" portainicio " component="luaMain"/>
    <media id="luaMain" src="main.lua"
descriptor="dsMain" />
  </body>
</ncl>
```

6.3 PROGRAMAÇÃO EM LUA

Será iniciado agora a criação do programa em linguagem Lua que será o “main.lua” chamado pela “media”.

Na barra do eclipse foi selecionado novamente o item “File”, e em seguida o item “New” e por último o ícone “New Lua File”.

Foi selecionado na janela “New Lua File Wizard” o “ProjetoNCLua” e então foi dado o nome para o arquivo Lua, que nesse artigo foi “main.lua”.

Após a conclusão foi selecionado o botão “Finish”

A eclipse criou o programa “main.lua” dentro da pasta do “ProjetoNCLua” ao lado do “NossoPrograma.ncl” Isto irá facilitar na hora da execução do exemplo.

Primeiro foi feito a declaração de duas variáveis do tipo local de nome “cw” e “ch”.

Essas variáveis receberam o valor devolvido pela função `canvas:attrSize()` que é uma função lua que retorna dois valores relativos ao tamanho da tela disponível. Esse valor foi determinado na região do programa NCL com o valor (`width="50%" height="50%"`) que quer dizer 50% do tamanho total da tela. Canvas é um módulo Lua que possibilita fazer operações gráficas. Segue a sintaxe da declaração:

```
local cw, ch = canvas:attrSize()
```

Para tornar bem visível a tela, foi escolhida a cor cinza, que é especificada na função `canvas:attrColor()`, e executada no desenho de um retângulo com a função `canvas:drawRect()` a qual recebeu o parâmetro “fill” que preenche toda a área e os parâmetros das variáveis do tamanho já explicadas. Os valores de coordenada X e Y do posicionamento foram mantidos em zero para preencher toda a área da tela encontrada, conforme o exemplo.

```
canvas:attrColor('silver')
canvas:drawRect("fill", 0, 0, cw, ch)
```

O texto foi colocado em azul repetindo a função `canvas:attrColor()` agora com o parâmetro “azul”, e a chamada da função `canvas:attrFont()` passando o parâmetro da fonte “vera” e o tamanho da fonte como “24”. Da seguinte forma:

```
canvas:attrColor("blue")
canvas:attrFont("vera", 24)
```

Para a exibição do texto foi criado uma variável tipo local de nome “texto” que recebeu o texto “Hello Word” e em seguida foi chamada a função `canvas:drawText()` que recebeu os parâmetro de “50” para distância da lateral, “60” para a distância do topo da tela e “texto” para o texto conforme sintaxe.

```
local texto = "Hello Word"
canvas:drawText(50, 60, texto)
```

Após isso foi chamada a função `canvas:flush()` que executa as funções `canvas`, da seguinte forma:

```
canvas:flush()
```

Terminando assim o aplicativo.

6.4 EXECUÇÃO DO PROGRAMA

Próximo passo foi conectar o SSH *Secure File Transfer* na máquina virtual.

A execução foi realizada copiando a pasta “ProjetoNCLua” de dentro da pasta “PrimeiroNCLua” para o diretório `misc/ncl30/` no SSH *Secure File Transfer*, seguindo a orientação do Virtual STB Ginga-NCL.

Foi executado o seguinte comando no SSH *Secure Shell*:

```
“/misc/launcher.sh/misc/ncl30/ProjetoNCLua/NossoPrograma.ncl”
```

O resultado foi o seguinte.

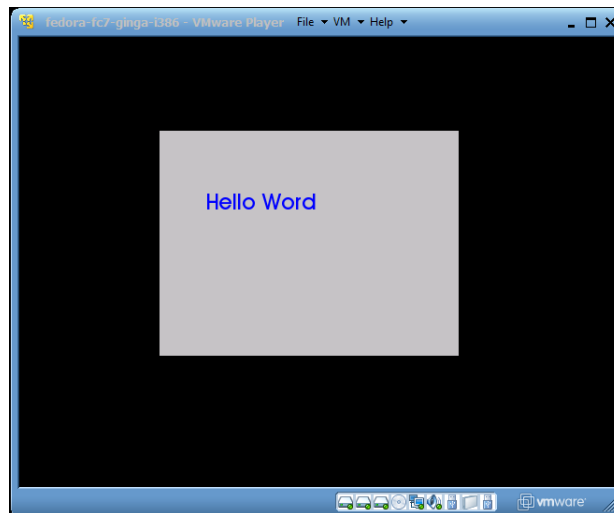


Figura 4: Execução do programa no STB.

Na execução, a aplicação permanece em funcionamento contínuo. Para finalizar a execução foi utilizado o comando “*killall ginga; fg 1*” no *software SSH Secure Shell*, ou a tecla (F11) na própria máquina virtual.

7. CONSIDERAÇÕES FINAIS

A tendência de toda tecnologia nova é que ela evolua com sua real aplicabilidade e, de fato, desde o seu lançamento até os dias atuais, muita coisa já evoluiu e está evoluindo, trabalho esse sendo dirigido pelas universidades brasileiras, destacando-se a iniciativa da PUC – Rio que está aperfeiçoando o Ginga-NCL e as linguagens Lua e NCL, o LAViD está trabalhando no Ginga-J, e vários profissionais da área de TI e Engenharia trabalhando no desenvolvimento das linguagens e criando aplicações.

Vale ressaltar que com essas escolhas, o Brasil passou a incentivar o desenvolvimento das ferramentas e tecnologias a ser utilizada na criação de aplicações para TV Digital Interativa, com um detalhe importantíssimo, todas elas sendo de padrão aberto, no que garante o contínuo aperfeiçoamento das mesmas.

8. REFERÊNCIAS

ABNT NBR 15606-2 (2007) – Associação Brasileira de Normas Técnicas, “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”, Sistema Brasileiro de TV Digital Terrestre, NBR 15606-2.

Fórum do Sistema Brasileiro de TV Digital Terrestre. [On Line] <http://www.forumsbtvd.org.br> - Visitado em maio de 2010.

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.

Portal do Software Público Brasileiro. [On Line] <http://www.softwarepublico.gov.br> – Visitado em junho de 2010.

The Programming Language Lua. [On Line] <http://www.lua.org/> - Visitado em maio de 2010.

Projeto TV Digital Interativa AEDB 2007. [On Line] <http://www.tvdigitaedb2007.blogspot.com/>