

# Ginga-J ou Ginga-NCL: características das linguagens de desenvolvimento de recursos interativos para a TV Digital

Airton Zancanaro<sup>1</sup>  
Paloma Maria Santos<sup>2</sup>  
José Leomar Todesco<sup>3</sup>

## Resumo

A televisão, como meio de comunicação de massa, segue uma tendência mundial de digitalização de sinal e conteúdo, incorporando a convergência de várias mídias. Três são os padrões de TV Digital existentes no mundo: Norte Americano (ATSC-T), Europeu (DVB-T) e Japonês (ISDB-T). O Brasil avaliou todos e optou pelo sistema Japonês. A adaptação deste à necessidade nacional, que faz uso da tecnologia produzida por brasileiros, fez surgir o Sistema Brasileiro de TV Digital Terrestre (SBTVD-T). Com a transmissão do sinal no formato digital, através da disponibilização de recursos interativos, há a intenção de proporcionar maior inclusão social do povo brasileiro. Este artigo apresenta uma abordagem do desenvolvimento de recursos interativos para a TV Digital, baseando-se no uso de dois paradigmas de programação, linguagens declarativas e procedurais, suportadas pelo *middleware* Ginga. *Middleware* este que, quando incorporado a um *set-top-box*, pode ser definido como a inteligência por traz da TV Digital. Este estudo se inicia a partir de uma revisão bibliográfica dos padrões de TV Digital existentes no mundo. Apresenta-se o conceito do recurso da interatividade, bem como seus níveis e sua importância dentro do sistema. Uma exploração é feita sobre o *middleware* Ginga, juntamente com os dois subsistemas lógicos, o Ginga-J, responsável por executar aplicações escritas na linguagem Java, e o Ginga-NCL, que executa as aplicações construídas na linguagem declarativa NCL. Neste estudo, buscou-se também uma comparação entre estas duas linguagens, tendo em vista apresentar características que visem ajudar a desenvolvedor, a escolher a linguagem mais adequada para construir aplicações interativas no Ginga. Nesta comparação, analisaram-se questões como a velocidade de processamento, acesso à documentação, popularidade, volume de código gerado, aderência segundo a norma que especifica esta questão, entre outros. Como resultado, buscou-se comparar as duas linguagens previstas para o desenvolvimento de aplicações interativas para o Ginga (Java e NCL), esclarecendo pontos e guiando os responsáveis por criar aplicativos, para que façam a melhor escolha. Assim, chega-se a conclusão que, pelo fato do Ginga-J ainda estar em fase de desenvolvimento, o melhor a fazer é construir aplicações usando NCL juntamente com a linguagem LUA. Espera da TV Digital que ela traga algo novo em termos de programação e aplicativos para o usuário, tendo em vista, a interação do telespectador com a emissora e com os conteúdos disponibilizados. Se esse processo não for amigável, a tecnologia com certeza será levada ao fracasso.

---

<sup>1</sup> Graduado em Ciências da Computação pela Universidade Regional de Blumenau. Mestrando do Programa de Pós-graduação em Engenharia e Gestão do Conhecimento pela Universidade Federal de Santa Catarina. Beneficiários de auxílio financeiro da CAPES-Brasil. Email: [airtonza@gmail.com](mailto:airtonza@gmail.com)

<sup>2</sup> Graduada em Engenharia de Telecomunicações pela Universidade Regional de Blumenau. Mestranda do Programa de Pós-graduação em Engenharia e Gestão do Conhecimento pela Universidade Federal de Santa Catarina. Beneficiários de auxílio financeiro da CAPES-Brasil. Email: [pmariasantos@yahoo.com.br](mailto:pmariasantos@yahoo.com.br)

<sup>3</sup> Graduado em Matemática pela Universidade Federal de Santa Catarina, Graduado em Educação Física pela Universidade do Estado de Santa Catarina, mestre e doutor em Engenharia de Produção também pela UFSC. Atualmente é Professor do curso de Sistemas de Informação na Graduação e do Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da UFSC. Pesquisador do Instituto Stela. Email: [tite@stela.org.br](mailto:tite@stela.org.br)

## **Palavras-Chave**

TV Digital. Ginga-NCL. Ginga-J. *Middleware*. Interatividade.

## **1. INTRODUÇÃO**

O dia 02 de dezembro de 2007 marcou o início de uma nova era na história das comunicações no Brasil. A televisão digital chega não apenas com o objetivo de ser a evolução tecnológica de um sistema, mas principalmente, com a perspectiva de trazer aos telespectadores recursos interativos nunca vistos anteriormente.

Além da melhoria na qualidade de áudio e vídeo, a televisão digital oferecerá aos usuários, conforme Crocomo (2007), a possibilidade de tirá-los da inércia a qual estavam submetidos.

Conectada a um terminal de acesso (*set-top-box*), à televisão digital permitirá que os telespectadores interajam com diversos programas e acessem inúmeras informações e serviços. Através da interatividade, conteúdos e serviços não disponíveis atualmente e sequer imaginados poderão ser disponibilizados. A esperança é de que o telespectador atue como protagonista desse novo processo.

A iniciativa do governo de não comprar a tecnologia já disponível em outros países e sim adaptá-la à realidade do povo brasileiro, fez com que instituições de ensino se mobilizassem por um objetivo comum. Temos hoje que o *Middleware* Ginga, juntamente com as linguagens NCL e Lua, construídas inteiramente no Brasil, estão se tornando um modelo para outros padrões existentes no mundo.

A seção 2 apresenta uma revisão bibliográfica sobre escolha do padrão brasileiro para a TV Digital. Na seção 3, apresenta-se o conceito do recurso da interatividade, bem como seus níveis e sua importância dentro do sistema. A seção 4 explora o *Middleware* Ginga e o especifica como a inteligência na TV Digital. Na seção 5, apresenta-se além de um panorama dos ambientes de programação disponíveis no Ginga, as arquiteturas Ginga-J e Ginga-NCL. A seção 7 traz um comparativo sobre essas linguagens e, finalmente, na seção 8, têm-se as conclusões deste trabalho.

## **2. ESCOLHA DO PADRÃO BRASILEIRO**

A escolha do padrão de televisão digital para implantação no Brasil surgiu como resultado da análise da viabilidade técnico-econômica de três padrões já consolidados em

outros países: o americano ATSC-T (*Advanced Television System Committee*), o europeu DVB-T (*Digital Video Broadcasting*) e o japonês ISDB-T (*Integrated Services Digital Broadcasting Terrestrial*). Depois da análise, concluiu-se que o padrão americano não poderia ser adotado principalmente pelo fato de não contemplar recursos relacionados à mobilidade e à portabilidade. Já o europeu, possuía sérias limitações quanto a disponibilizar, simultaneamente, qualidade de imagem na recepção fixa e portabilidade. Esta última, quando disponível, era passível de cobrança de taxa de utilização. Apesar disso, o padrão europeu se mostrou bastante flexível e com penetração maior do que o padrão americano. O padrão japonês apresentou maiores vantagens em relação aos concorrentes. Ele se configurou como o mais avançado e o que oferecia maior convergência e possibilidade de segmentação. Ao mesmo tempo em que o padrão oferecia alta definição, permitia portabilidade, de forma gratuita e simultânea. Ainda, o padrão japonês dava suporte para que Brasil construísse o seu próprio *middleware*: o Ginga, desenvolvido livremente, em plataformas abertas.

Dentre outros atributos, pesou também a favor da escolha pelo modelo japonês como base, a possibilidade de implementação de recursos interativos. Aqui, sem dúvida, reside a maior expectativa de retorno desse novo sistema, foco deste artigo.

O ISDB-TB (*International System for Digital Broadcasting-Terrestrial*), adaptado do sistema Japonês (ISDB-T), e também denominado de sistema Nipo-Brasileiro, se configurou como o padrão de transmissão de TV Digital no Brasil, tendo como vantagem perante os outros modelos além do uso de um *middleware* exclusivo, a adoção de um padrão de compressão de vídeo muito mais moderno e eficiente: o MPEG-4 ou H.264, conforme observa Mizukosi (2008), além do AAC para compressão de áudio.

Na sequência, abordar-se-á o conceito e a classificação da interatividade em seus três níveis.

### **3. RECURSO DA INTERATIVIDADE**

A interatividade tem como princípio básico transformar o papel do receptor. Com a interatividade, a TV deixa de ser unidirecional e passa a não apenas mandar informações, mas também a receber.

“O telespectador passa a ter um canal para se comunicar com a emissora, tirando-o da inércia a qual está submetido (...). O grau dessa interatividade vai depender dos serviços oferecidos e, principalmente, da velocidade do canal de retorno” (MONTEZ e BECKER, 2004, p.13).

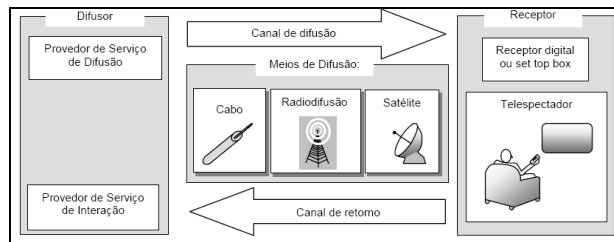


Figura 1: Modelo de um sistema de televisão digital Interativa  
 Fonte: BECKER e MONTEZ, 2005.

O canal de retorno, segundo Montez e Becker (2004), é esse meio através do qual é possível a troca de informações no sentido inverso da difusão, ou seja, do telespectador para a emissora.

Essa interatividade pode se dar segundo três níveis, conforme aborda Fernando Crocomo (2007). A interatividade local (“nível 1”) não necessita de um canal de retorno e está relacionada aos conteúdos transmitidos e armazenados no *set-top-box*, através do qual o usuário pode navegar entre as opções disponíveis. A interatividade “nível 2” requer a existência de um canal de retorno, via rede telefônica ou internet, onde é possível retornar a mensagem, não necessariamente em tempo real. Na interatividade “nível 3”, o canal de retorno fica sempre ativo e é possível enviar e receber informações em tempo real.

A interatividade, sem dúvida, é o grande trunfo da TV Digital. Ela é a chave para o acesso dos, até então, meros receptores ao mundo da produção e do compartilhamento de conteúdo, através da televisão. É a interatividade que vai permitir que os telespectadores, enfim, façam parte de uma rede colaborativa de construção do conhecimento.

Em seguida, apresentar-se-á o *middleware* Ginga como sendo a inteligência por trás da TV Digital.

#### 4. INTELIGÊNCIA DA TV DIGITAL

A inteligência por trás da nova tecnologia digital está no chamado *middleware* Ginga. Ele é processado a partir do *set-top-box* com a função de suportar as aplicações que são transmitidas junto com a programação recebida. O Ginga faz o “meio de campo” entre os aplicativos interativos e o sistema operacional Linux.

De acordo com a *International System for Digital TV* (ISDTV, 2007), o Ginga permite que se desenvolvam aplicações interativas para a TV Digital, independentemente da plataforma de hardware dos fabricantes dos *set-top-boxes*.

Como benefícios da utilização do *middleware* Ginga para a cadeia de radiodifusores, o ISDTV (2007) cita:

- Viabilização de novas ações interativas;
- Utilização do canal de retorno, viabilizando interações remotas com usuários, fortalecendo possíveis parcerias com as operadoras e aumentando a receita;
- Portabilidade das aplicações para outros padrões de TV Digital, facilitando a exportação de programas para mercados internacionais.

Ainda, benefícios para os que utilizam o Ginga também percorrem a cadeia dos desenvolvedores. Segundo o ISDTV (2007) tem-se:

- Integração com vários sistemas operacionais, linguagens *script* e linguagens de sincronização de mídia;
- Capacitação de empresas regionais exportarem soluções e serviços baseados em TV Digital;
- Possibilidade de desenvolver aplicativos com acesso aos recursos do aparelho, independente do fabricante;
- Reutilização de componentes de software com redução de custos no desenvolvimento de novos aplicativos.

Desenvolvido pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em conjunto com a Universidade Federal da Paraíba (UFPB), o sistema Ginga é subdividido em dois subsistemas principais interligados, que, segundo informações do *site* [www.ginga.org.br](http://www.ginga.org.br), permitem o desenvolvimento de aplicações seguindo dois paradigmas de programação diferentes.

Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma será mais adequado que o outro. Esses dois subsistemas são chamados de Ginga-NCL (para aplicações declarativas em NCL) e Ginga-J (para aplicações procedurais em Java). Na Figura 2 está representado o modelo estrutural do Ginga.

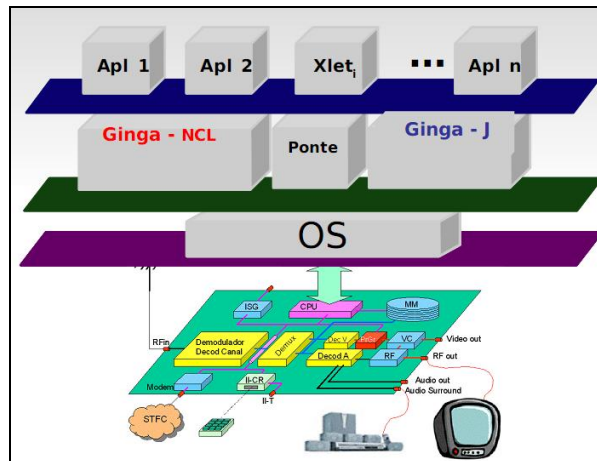


Figura 2 – Middleware Ginga  
Fonte: BURLAMAQUI (2009)

Na sequência, apresentam-se os dois paradigmas de programação, a procedural e a declarativa, suportados pelo *middleware* Ginga.

## 5. AMBIENTES DE PROGRAMAÇÃO

Para Soares (2007), o “*middleware* é uma camada de software localizada entre as aplicações (programa de uso final) e o sistema operacional”. Ele tem como objetivo esconder detalhes das camadas inferiores e a unificação dos diferentes tipos hardwares e sistemas operacionais, fazendo com que os que produzem conteúdo tenham uma visão única.

Normalmente, nos ambientes de desenvolvimento para TV Digital usam-se dois paradigmas de programação: declarativas e procedurais.

As linguagens declarativas seguem um alto nível de abstração, normalmente ligada a um domínio ou objetivo específico. Neste modelo, o programador fornece um conjunto de tarefas que deverão ser executadas, não se preocupando com os detalhes de como o interpretador, compilador, máquina real ou virtual fará este procedimento. Dentre as linguagens declarativas mais comuns para a TV Digital, pode-se citar a NCL e o XHTML (BML-ARIB, DVB-HTML e ACAP-X).

Por outro lado, têm-se as linguagens procedurais onde o programador tem maior poder sobre o código, informando cada passo que será executado. Para isso, é necessária uma expertise maior por parte do profissional. Dentre tantas existentes, o Java é a linguagem mais usada para desenvolver sistemas nesta modalidade de programação para a TV Digital (BARBOSA; SOARES, 2008).

## 5.1 GINGA-J

Para DTV - TV DIGITAL BRASILEIRA (2009), “O Ginga-J foi desenvolvido pela UFPB para prover uma infraestrutura de execução de aplicações baseadas na linguagem Java, com facilidades especificamente voltadas para o ambiente de TV digital”.

O Ginga-J dá suporte à linguagem de programação procedural Java, composta por APIs (*Application Programming Interface*) que tem o objetivo de suprir as funcionalidades necessárias para a implementação de aplicativos usados na TV Digital. Com ela, é possível manipular vídeos, áudios, textos e até protocolos de acesso (GINGA-J... 2009).

Segundo Soares (2007) Ginga-J “[...] é o subsistema lógico do middleware Ginga responsável pelo processamento de aplicações imperativas escritas utilizando a linguagem Java”.

A arquitetura do Ginga-J deverá ser compatível com as especificações da norma da ABNT (Associação Brasileira de Normas Técnicas), NBr 15606-4, que será publicada em breve. Suas APIs são divididas em três módulos, segundo Soares (2008), e conforme descrito na Figura 3:

- As APIs verdes, do núcleo, são específicas e responsáveis por manter o máximo possível de compatibilidade com os sistemas: europeu e americano. Inclui o pacote Java DTV;
- As APIs amarelas são inovações brasileiras que também podem ser exportadas para outros sistemas. Elas oferecem suporte aos múltiplos usuários, a múltiplos dispositivos e a múltiplas redes. Também oferecem suporte para aplicações que poderão ser recebidas, armazenadas e executadas no futuro. Esta API inclui o JMF (*Java Media Framework*) que é usada para desenvolver aplicações avançadas como a captura de áudio e vídeo, entre outras;
- As APIs vermelhas dão suporte às aplicações voltadas para o Brasil, especialmente para a interatividade, promovendo a inclusão social. Permitem também a integração do conteúdo declarativo e procedural na mesma aplicação.

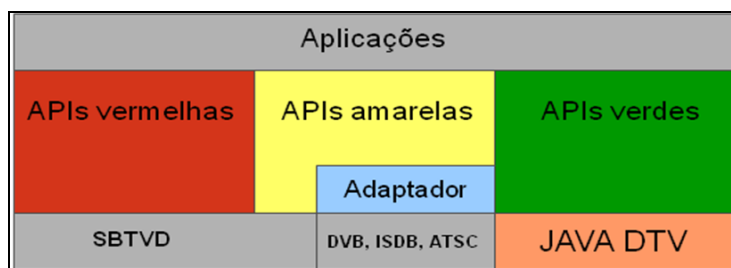


Figura 3 – APIs – Ginga J

Java DTV é um conjunto de APIs, conforme mostrado na Figura 4, e pode ser considerado como o principal pacote do Ginga-J. Ele foi criado pela Sun Microsystem, em parceria com o Fórum SBTVD (Sistema Brasileiro de TV Digital), em substituição ao GEM (*Globally Executable MHP*). Este possui pacotes que não são *Royalties Free* e, por este motivo, não se enquadram nos objetivos do Fórum.

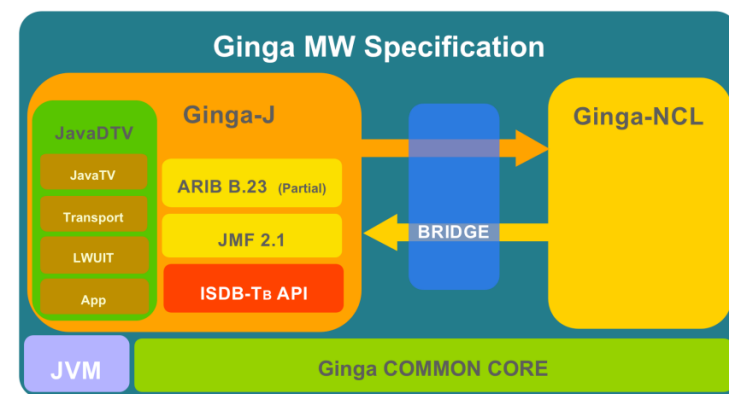


Figura 4 – Modelos Ginga  
 Fonte: KULESZA; FERREIRA (2009)

O GEM são APIs usadas nos *middlewares* dos padrões americano, europeu e japonês. Nele, encontram-se os seguintes pacotes:

#### *JavaTV*

A linguagem JavaTV tem como principal objetivo o desenvolvimento de aplicações com alto nível de interatividade, gráficos de qualidade e o processamento, que poderá ser efetuado dentro do *set-top-box*. Este processamento é possível desde que o equipamento tenha o interpretador de *bytecodes*, que é a Máquina Virtual Java instalada.

#### *Davic*

É um conjunto de especificações que tem por objetivo além de sustentar a interoperabilidade real de ponta a ponta na execução de serviços de áudio e vídeo, transmitidos via radiodifusão, também o de permitir a interatividade com o usuário.

#### *Havi*

Este pacote está focado na interoperabilidade de dispositivos de áudio e vídeo, de forma que possam interagir uns com os outros. O objetivo principal é permitir que o usuário tenha o controle desses componentes de forma familiar, usando o controle remoto ou outro meio de interação.



## 5.2 GINGA-NCL

A partir deste ponto, descreve-se o Ginga-NCL de maneira a dar a ideia para usuários leigos do que se trata e o que é possível construir utilizando esta linguagem de programação no contexto da TV Digital.

Para Soares (2007) Ginga-NCL é “[...] o subsistema lógico do *middleware* Ginga responsável pelo processamento de aplicações declarativas NCL”.

A linguagem de programação NCL (*Nested Context Language*) foi desenvolvida pela PUC-Rio com o objetivo de facilitar as especificações de interatividade, sincronismo espaço-tempo entre os objetos de mídia, adaptabilidade, suportar múltiplos dispositivos e suportar programas ao vivo interativos não-lineares (DTV - TV DIGITAL BRASILEIRA, 2009).

Baseada no modelo NCM (*Nested Context Model*), e seguindo os princípios adotados pelo W3C (*World Wide Web Consortium*), a NCL é uma linguagem XML<sup>4</sup>, declarativa e que mantém os objetos semanticamente unidos em uma apresentação de multimídia.

Para Soares (2007), “Ginga-NCL é o subsistema ‘Lógico’ do *middleware* Ginga que processa documentos NCL”. O Formatador NCL recebe um documento NCL e controla a sua apresentação, sincronizando os objetos de mídia, fazendo com que elas sejam apresentadas no momento programado.

De acordo com Barbosa e Soares (2008, p.18), “um aplicativo NCL apenas define como os objetos de mídia são estruturados e relacionados no tempo e espaço”, formando assim uma linguagem de “cola”.

Soares Neto et al. (2009) dizem que para construir documentos de hipermídia, é necessário responder algumas perguntas: **O** que se quer tocar? **Onde** mostrar na tela? **Como** tocar (volume e com que *player*)? **Quando** tocar (antes/depois de qual mídia a ser apresentada ou após pressionar determinada tecla)?

### *O que tocar*

Normalmente, quando se planeja um programa áudio visual interativo, pensa-se primeiro no conteúdo. Os nós de mídia representam este conteúdo e todos são definidos dentro de um contexto (<body>).

---

<sup>4</sup> XML, segundo Alecrin (2009), “[...] é uma especificação técnica desenvolvida pela W3C (*World Wide Web Consortium* - entidade responsável pela definição da área gráfica da internet), para superar as limitações do HTML, que é o padrão das páginas da Web”.

### *Onde Mostrar*

Conforme o conteúdo está sendo concebido, é necessário também definir o local (regiões) onde ele será apresentado na tela do expectador. Assim, uma região delimita onde as mídias poderão ser apresentadas.

### *Como tocar*

Neste momento, os descritores fazem a associação da mídia com a região. Os descritores servem também para definir como a mídia deverá ser apresentada.

### *Quando tocar*

Neste caso, para o primeiro nó, é necessário criar uma porta no contexto *body*. Caso haja mais de uma porta criada, elas serão iniciadas em paralelo.

Antes da apresentação da estrutura de um documento NCL, vale ressaltar que, para facilitar o trabalho do programador, no momento de criar o aplicativo, o ideal é utilizar um editor específico para a linguagem. O Eclipse é um desses editores, usado para várias linguagens de programação. No caso da NCL, o usuário deverá instalar um *plugin* que lhe dará as funcionalidades necessárias.

O *download* do editor Eclipse pode ser efetuado a partir do *site* [www.eclipse.org](http://www.eclipse.org). Já o *plugin* e seu manual de instalação, podem ser encontrados nos seguintes endereços:

<http://laws.deinf.ufma.br/~ncleclipse/> e

[http://laws.deinf.ufma.br/~ncleclipse/doc/tutoNCLEclipse\\_1.0.0.pdf](http://laws.deinf.ufma.br/~ncleclipse/doc/tutoNCLEclipse_1.0.0.pdf).

A PUC-Rio também desenvolveu um editor denominado de Composer, tendo como finalidade a construção de arquivos NCL de forma visual. Neste caso, os usuários não necessitam conhecer a linguagem para construir programas interativos. Este sistema pode ser conseguido através do endereço: <http://www.ncl.org.br/ferramentas.php>.

De posse do Eclipse, pode-se observar que a NCL é muito similar à codificação HTML, que possui um cabeçalho, um corpo de documento e onde toda a declaração deve ser aberta com uma *tag* `<>` e fechada com `</>`. Na tabela 1, pode-se observar uma estrutura básica de um documento NCL.

Tabela 1 – Estrutura básica de um documento NCL

<b>Cabeçalho do arquivo NCL</b>	1: <code>&lt;!-- Início do cabeçalho do documento --&gt;</code> 2: <code>&lt;?xml version="1.0" encoding="ISO-8859-1"?&gt;</code> 3: <code>&lt;ncl id="exemplo01" xmlns="http://www.telemidia.puc-rio.br/specs/xml/NCL23/profiles" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</code>
---------------------------------	---

	xsi:schemaLocation="http://www.telemidia.puc-rio.br/specs/xml/NCL23/profiles/NCL23.xsd">
<b>Cabeçalho do programa</b>	4: <head>
Base de regiões	5: <regionBase> 6: <!-- regiões da tela onde as mídias são apresentadas --> 7: </regionBase>
Base de descritores	8: <descriptorBase> 9: <!-- descritores que definem como as mídias são apresentadas --> 10: </descriptorBase>
Base de conectores	11: <connectorBase> 12: <!-- conectores que definem como os elos são ativados e o que eles disparam --> 13: </connectorBase>
	14: </head>
<b>Corpo do programa</b>	15: <body>
Ponto de entrada no programa	16: <!-- Para ser exibido o primeiro nó deve ser criada porta --> 17: <port id="pInicio" component="ncPrincipal" interface="iInicio"/>
Conteúdo do programa	18: <!-- contextos, nós de mídia, elos e outros elementos -->
	19: </body>
<b>Término do programa</b>	20: </ncl>

Fonte: Soares Neto et al. (2009)

Para exemplificar a construção de um documento NCL, a Figura 4 representa a visão de *layout*, onde o vídeo deverá ser exibido no canto superior esquerdo da tela do espectador e executado no momento que a aplicação for iniciada.

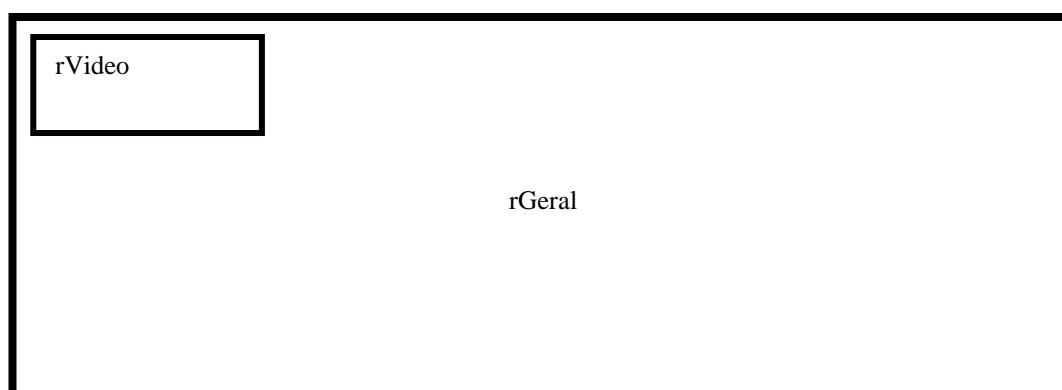


Figura 4 – Visão de *layout* do exemplo

No Quadro 01 é apresentado o *script* da aplicação e, abaixo, na Figura 5, a imagem da execução do mesmo.

Quadro 01 – Exemplo de aplicativo NCL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="exemplo02" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
```

```

<head>
  <regionBase>
    <!-- regiões/área da tela onde as mídias serão exibidas -->
    <region id="rGeral" width="100%" height="100%">
      <region id="rVideo" left="10" top="10" height="480" width="640"/>
    </region>
  </regionBase>
  <descriptorBase>
    <!-- como as mídias serão exibidas -->
    <descriptor id="dVideo" region="rVideo"/>
  </descriptorBase>
</head>
  <!-- início do corpo do programa -->
<body>
  <!-- define a porta de entrada do programa, apontando para o primeiro nó a ser exibido -->
  <port id="pVideo" component="mVideo"/>
  <!-- início da definição das mídias que são usadas no programa -->
  <media id="mVideo" descriptor="dVideo" src="midia/video1.mpg" type="video/mpeg">
  </media>
</body>
</ncl>

```



Figura 5 – Execução no *script* no NCL-Player

A NCL permite adicionar objetos de mídia baseados em XHTML<sup>5</sup>, mas não embute documentos ou objetos XHTML. Ela tem apenas o suporte do formatador para exibir estes arquivos.

Além do XHTML, outra linguagem que a NCL dá suporte é a LUA: uma linguagem em *script* que combina sintaxe procedural com a declarativa. Escrita em C, ela é simples, leve, robusta, embarcada e de código aberto. Por sua natureza extensível, ela trabalha embarcada em outro programa chamado de hospedeiro ou, programa principal, no caso NCL. O

<sup>5</sup> XHTML (*Extensible Hyper Text Markup Language*) foi criado com o objetivo de ser uma linguagem em conformidade com o XML e ao mesmo tempo compatível com o HTML.

programa principal tem a função de chamar trechos do código LUA, escrever e ler variáveis, entre outras funções (IERUSALIMSKY, 2003).

Aplicações construídas em NCL em conjunto com a LUA, aumentam muito o potencial dos programas para TV Digital, levando a um maior grau de imersão e interatividade a serem disponibilizadas para o usuário.

Observam-se algumas telas de programas interativos para a televisão, retirados do site <http://clubencl.org>, que foram produzidos usando as linguagens comentadas anteriormente.

Como primeiro exemplo, na Figura 6, tem-se o programa “Viva Mais”, que foi o primeiro produzido por brasileiros com o objetivo de interatividade. Nesta aplicação foram utilizadas as linguagens LUA e NCL.



Figura 6- Viva Mais – Peso ideal

Na Figura 7, é apresentada a primeira série de comerciais criadas pela empresa Proview Produção para ilustrar o espaço interativo. Esta aplicação foi construída inteiramente usando o Ginga-NCL.



Figura 7 – Comercial Proview I

Apesar de a NCL ser uma linguagem de marcação e possuir algumas restrições, quando associada a outra linguagem de natureza procedural, ocorre um casamento perfeito, sendo possível a solução de problemas de forma rápida e facilitada.

Na sequência, comparam-se as arquiteturas Ginga-J através da linguagem Java e do Ginga-NCL através da linguagem NCL.

## 6. GINGA-J VERSUS GINGA-NCL

Um comparativo entre linguagens sempre gera muita polêmica, principalmente quando não há comprovação bibliográfica sobre o que está sendo afirmado. A validação fica restrita à interpretação do pesquisador a respeito do pouco que já foi testado e do muito que já foi especulado. Quando a linguagem está em pleno desenvolvimento, essa comprovação se torna ainda mais complicada.

Levando isso em consideração, se traz, como contribuição final deste artigo, um comparativo entre as linguagens Java e NCL, com o intuito de mostrar que “excluir” uma linguagem de programação para implementação em determinado sistema pelo fato da mesma exigir pagamento de licença, nem sempre é o melhor caminho. Muitas vezes, o ganho que se tem sob as perspectivas de produtividade, acesso à documentação, velocidade de processamento, entre outras, supera qualquer limitação meramente financeira.

Tomando como parâmetro o tamanho da comunidade de desenvolvimento no âmbito mundial, a linguagem de programação Java, do Ginga-J, possui uma infinidade de programadores maior do que o da NCL.

Quando se fala em tamanho de código gerado em uma aplicação, a NCL, devido a suas características declarativas, possui menor volume de código gerado do que a Linguagem Java. Por outro lado, o Java permite um maior controle do programador sobre o código.

Em relação à produtividade, a NCL também ganha do Java, devido a sua facilidade de programação. Em se tratando da velocidade de processamento, a linguagem Java é mais robusta, necessitando de mais capacidade de processamento e, portanto, sendo mais lenta do que a linguagem NCL.

Devido ao uso de forma extensiva da linguagem Java, o acesso à documentação (Javadoc) é facilitado em detrimento da NCL, onde os registros ficam restritos a quem desenvolve para a TV Digital.

Por muito tempo, o Brasil vem sendo um consumidor de tecnologia na área de televisão, isto é, compra os produtos prontos e paga caro. Com o projeto de implantação da TV Digital, deu-se a chance aos pesquisadores brasileiros de pôrem em prática as pesquisas realizadas nas universidades. Neste caso, NCL e LUA são linguagens genuinamente brasileiras. Isto já não acontece com o Java, que é produzido pela multinacional Sun Microsystems, sendo que a mesma possui negócios em mais de 100 países ao redor do mundo.

O OpenGinga é a plataforma de desenvolvimento do Gingga-J, e está em fase de construção. Levará ainda algum tempo para que se tenha todas as implementações, juntamente com a estabilidade necessária para a construção de aplicações em larga escala.

## **6.1 NORMA ABNT NBR 15602**

A norma ABNT 15602 especifica o conjunto de funcionalidades essenciais requeridas dos dispositivos de recepção de televisão digital, destinados a receber sinais na modalidade fixa, móvel e portátil.

De acordo com esta norma, as aplicações construídas em NCL e LUA obrigatoriamente deverão funcionar tanto para celular (equipamento móvel) quanto para *set-top-box*. É válido notar que a presença do Java em aparelhos celulares é opcional, ficando a critério, do fabricante, incorporar ou não a tecnologia nos receptores móveis.

A Tabela 2 apresenta uma parametrização da presença das linguagens Java e NCL, relacionando os tipos de receptores disponíveis para captura do sinal digital de televisão, de acordo com a norma.

Funcionalidades		Tipo de Receptor	
		Set-Top-Box	Celular
<b>Linguagem de Programação</b>			
Ginga	NCL	Obrigatório	Obrigatório
	Java	Obrigatório	Opcional
<b>Máquina de Execução</b>			
Engine	Máquina Virtual Java	Obrigatório	Opcional
	LUA	Obrigatório	Obrigatório
<b>Ponte entre as Linguagens</b>			
Ponte	LUA	Obrigatório	Opcional
	ECMAScrip	Opcional	Opcional

A normatização do Ginga-NCL está pronta, conforme as normas ABNT NBr 15602-2:2007 e 15602-5:2008, que tratam respectivamente: na Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações; e na Parte 5: Ginga-NCL para receptores de transmissão para receptores portáteis – Linguagem XML para codificação de aplicação. Já, em relação, a norma do Ginga-J será publicada em breve.

Assim, para que o desenvolvedor de conteúdo interativo para a TV Digital não se depare com dispositivos móveis que não estejam preparados para o Java, e assim correr o risco de sua aplicação não funcionar e ficar fadado ao descaso, o uso da NCL, juntamente, com a linguagem LUA se faz necessário.

## 7. CONCLUSÃO

O objetivo deste artigo foi o de apresentar o *middleware* Ginga como um ambiente que possibilita o desenvolvimento de aplicativos interativos para a TV Digital, a partir de diversas linguagens de programação.

Procurou-se mostrar uma comparação entre as linguagens Ginga-J e Ginga-NCL, das quais o *middleware* irá utilizar para disponibilizar a tão esperada interatividade.

Através da comparação, e tendo como base o exposto na norma ABNT NBR 15604, tem-se que, para garantir que as aplicações interativas a serem disponibilizadas para a televisão digital possam ser acessadas pelo usuário independentemente do tipo de equipamento de recepção que este tenha, essas devem ser desenvolvidas usando a linguagem



declarativa NCL. Desta forma, os desenvolvedores não ficariam presos “na mão” dos fabricantes de celular, sem saber se o Java está ou não disponível no aparelho, visto que, a norma estabelece a não obrigatoriedade desta linguagem nos aparelhos móveis.

Buscou-se salientar ainda que, nem sempre a escolha da linguagem deve estar vinculada somente ao aspecto financeiro. Outras características devem ser analisadas, tendo em vista principalmente, a facilidade de uso do sistema por parte do telespectador, que é o maior interessado no processo.

Ambas as linguagens analisadas possuem suas vantagens e desvantagens. Cabe, portanto, ao programador conhecer as potencialidades de cada uma delas e adequar a mais apropriada a sua necessidade.

Por fim, o que se espera da TV Digital é que ela traga algo novo em termos de programação e aplicativos para o usuário, tendo em vista, a interação do telespectador com a emissora e com os conteúdos disponibilizados. Se esse processo não for amigável, a tecnologia com certeza será levada ao fracasso.

## **AGRADECIMENTO**

Agradecemos à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo financiamento das nossas pesquisas. Agradecemos também, ao professor do programa de Engenharia e Gestão do Conhecimento (EGC) da Universidade Federal de Santa Catarina, José Leomar Todesco, pelo empenho e dedicação em guiar os autores na confecção deste trabalho.

## **REFERÊNCIAS**

ALECRIM, Emerson. **Linguagem XML**. Publicado em 30/06/2003. Disponível em: <<http://www.infowester.com/lingxml.php>>. Acesso em: 24 jul. 2009.

BARBOSA S.D.J.; SOARES, L.F.G. **Tv Digital interativa no Brasil se faz com Ginga: Fundamentos, Padrões, Autoria Declarativa e Usabilidade**. Em T. Kowaltowski & K.Breitman (orgs.) Atualizações em Informática 2008. Rio de Janeiro, RJ: Editora PUC-Rio, 2008. pp.105-174

BECKER, V. e MONTEZ, C. **TV Digital Interativa: Conceitos, desafios e perspectivas para o Brasil**. Florianópolis: Editora da UFSC, 2005.

BURLAMAQUI, Aquiles. **Construindo programas de TV Digital Interativa usando NCL e Java**. Disponível em: <[http://gingarn.wdfiles.com/local--files/start/TVDI\(gingaj&gingancl\).pdf](http://gingarn.wdfiles.com/local--files/start/TVDI(gingaj&gingancl).pdf)>. Acesso em: 15 maio 2009.

CROCOMO, F. A. **TV digital e produção interativa: a comunidade manda notícias**. Florianópolis: UFSC, 2007. 178p.

DTV - TV DIGITAL BRASILEIRA. **O que é Ginga?** Disponível em: <<http://www.dtv.org.br/materias.asp?menuid=3&id=6>>. Acesso em: 12 maio 2009.

GINGA-J Disponível em: <<http://gingarn.wikidot.com/gingaj>>. Acesso em: 14 maio 2009.

IERUSALIMSCHY, Roberto. **Programming in Lua**. Rio de Janeiro: Ed. do Autor, 2003. xviii, 268 p.

**ISDTV: International System for Digital Television (2007) – Uma opção regional**. Disponível em <<http://www.senado.gov.br/sf/senado/unilegis/Curso%20TVDIGITAL/Meterial%20didatico%20%20semana/ISDTV%20Vers%C3%A3o%20002.ppt>>. Acesso em: 14 maio 2009.

KULESZA, Raoni; FERREIRA, Jefferson. **Desenvolvimento Ginga-J: JavaDTV - Open Ginga**. Disponível em: <<http://graphs.ucpel.tche.br/tvdi/OficinaGingaJ-Parte1.pdf>>. Acesso em: 22 jul. 2009.

MIZUKOSI, D. M. **A TV Digital Interativa – Definições e perspectivas**. Disponível em <<http://www.metodista.br/lato/tv-interativa/artigos/DanielMizukosi.pdf>>. Acesso em: 12 maio 2009.

MONTEZ, C.; BECKER, V. **TV Digital Interativa: Conceitos e Tecnologias**. In: WebMidia e LA-Web 2004 – Joint Conference. Ribeirão Preto, SP, Outubro de 2004.

SOARES, Luiz F. G. **Ambiente para desenvolvimento de aplicações declarativas para a TV digital brasileira**. TV digital: qualidade e interatividade / IEL.NC.– Brasília : IEL/NC, 2007.

SOARES, Luiz F. G. . As Múltiplas Possibilidades do Middleware Ginga. Produção Profissional: **Revista de Comunicação e Técnica Audiovisual**, São Paulo, p. 76 - 83, 01 jun. 2008.

SOARES, Luiz F. G. . **TV Interativa se Faz com Ginga**. Revista da SET-Sociedade Brasileira de Engenharia de Televisão, p. 30 - 35.

SOARES NETO, Carlos Salles et al. **Construindo Programas Audiovisuais Interativos Utilizando a NCL 3.0 e a Ferramenta Composer**. Disponível em: <<http://www.ncl.org.br/documentos/TutorialNCL3.0-2ed.pdf>>. Acesso em: 24 jul. 2009.