

Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnologia  
Curso de Ciência da Computação

PABLO ROBERT PEREIRA ALVES

***creaTiVe - PROPOSTA DE FERRAMENTA DE AUTORIA  
PARA TV DIGITAL VOLTADA AO USUÁRIO FINAL***

São Luís  
2012

**PABLO ROBERT PEREIRA ALVES**

***creaTiVe - PROPOSTA DE FERRAMENTA DE AUTORIA  
PARA TV DIGITAL VOLTADA AO USUÁRIO FINAL***

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Carlos de Salles Soares Neto.

São Luís

2012

Alves, Pablo Robert Pereira

creaTiVe - Proposta de Ferramenta de Autoria para TV Digital voltada ao usuário final / Pablo Robert Pereira Alves. - São Luís, 2012

47f.

Impresso por computador (Fotocópia).

Orientador: Carlos de Salles Soares Neto.

Monografia (Graduação) - Universidade Federal do Maranhão, Curso de Ciência da Computação, 2012.

1. TV Digital 2. Web 3. Interatividade 4. Ginga-NCL  
5. Ferramenta de desenvolvimento. I. Título.

CDU 004.738.5:654.19

PABLO ROBERT PEREIRA ALVES

***creaTiVe - PROPOSTA DE FERRAMENTA DE AUTORIA  
PARA TV DIGITAL VOLTADA AO USUÁRIO FINAL***

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 25 de junho de 2012

**BANCA EXAMINADORA**

---

Carlos de Salles soares Neto

Doutor em Computação

---

Mario Meireles Teixeira

Doutor em Computação

---

Maria Auxiliadora Freire

Mestre em Engenharia Civil

---

Roberto Gerson Albuquerque Azevedo

Mestre em Informática

*A todos que estiveram presente nessa fase decisiva e também àqueles que já se foram.*

## Agradecimentos

Aos meus pais, pois me concederam o privilégio de vir ao mundo e fizeram de tudo pela minha educação e criação.

A Davih, meu filho, porque é minha vida; e meu orgulho.

À minha namorada, Luciana, por ter estado ao meu lado durante toda a jornada.

Aos meus irmãos pela companhia e pela força que sempre me deram.

Ao meu orientador, que achou que esse projeto seria possível e me apoiou, me guiando da melhor maneira.

Ao Sonic, Mário, Alis e Link pela infância inesquecível.

Ao meu violão, companheiro inseparável.

A todos que acreditaram que eu chegaria até aqui.

*“Para ter sucesso, precisa-se de amigos. Para ter muito sucesso, precisa-se de inimigos”*

*Sidney Sheldon*

## RESUMO

A característica interativa da TV digital ainda caminha a passos lentos atualmente, mas isso está para mudar. Com a chegada de televisores com Ginga embarcado no mercado nacional e com a possibilidade da extinção do modo de transmissão atual de TV, a demanda por ferramentas que auxiliem a autoria de aplicações voltadas para essa plataforma começa a aumentar de forma perceptível. A proposta deste trabalho é fornecer um meio de alavancar a autoria de aplicações interativas NCL por meio de uma interface amigável para usuários não programadores através do navegador.

Palavras-chave: TV Digital. Web. Interatividade. Ginga-NCL. Ferramentas de Desenvolvimento.

## Abstract

Digital TV interactivity feature still walks slowly nowadays but this is about to change. As TVs with embedded Ginga middleware started to pop up in national market and with the possibility of extinction of current TV transmission mode, demand for aid authoring tools for this platform begin to noticeably grow up. The goal of this project is to provide a way to leverage authoring NCL interactive applications by using of user-friendly interface for non-programmer users through web browser.

Key-words: Digital TV. Web. Interactivity. Ginga-NCL. Development Tools.

## Lista de Tabelas

3.1	Comparativo das ferramentas descritas . . . . .	29
6.1	Situação final dos requisitos levantados . . . . .	45

## Lista de Figuras

2.1	Exemplo de arquivo XML . . . . .	16
2.2	Orientação a Objetos em PHP . . . . .	19
3.1	NCL Eclipse - visualização de Mídia e de Região . . . . .	26
3.2	Novo Composer mostrando diversas visualizações simultaneamente . .	27
3.3	Ferramenta TVision . . . . .	28
3.4	Tvision - Problemas na utilização . . . . .	28
4.1	Trecho do arquivo de tradução pt-br . . . . .	32
4.2	Exemplo de arquivo TIIF extraído do template “vivamais” . . . . .	34
5.1	Tela inicial da ferramenta . . . . .	35
5.2	Barra de ferramentas . . . . .	35
5.3	Aplicações NCL originais . . . . .	36
5.4	Criando um novo projeto . . . . .	36
5.5	Adicionando arquivos . . . . .	37
5.6	Árvore de Projeto . . . . .	37
5.7	Visualizador de Imagem . . . . .	38
5.8	Tocador de vídeo . . . . .	38
5.9	Propriedades de mídia e região . . . . .	39
5.10	Mídias do template “vivamais” . . . . .	40
5.11	Aplicação “Qual a cidade?” exibindo a resposta correta . . . . .	40
5.12	Aplicação “Álbum de São Luís” sendo exibida . . . . .	42

## Lista de Siglas

- AAC** - Advanced Audio Coding (Codificação avançada de áudio)
- ABNT** - Associação Brasileira de Normas Técnicas
- AJAX** - Asynchronous Javascript and XML (Javascript e XML Assíncrono)
- API** - Application Programming Interface (Interface para Programação de Aplicações)
- CSS** - Cascade Stylesheet
- DOM** - Document Object Model (Modelo de Objeto de Documento)
- HD** - High Definition (Alta definição)
- HTML** - Hypertext Markup Language (Linguagem de Marcação de Hipertexto)
- IDE** - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
- JSON** - Javascript Object Notation (Notação de Objeto Javascript)
- MPEG4** - Motion Picture Experts Group-4
- NCL** - Nested Context Language (Linguagem de Contexto Aninhada)
- OOP** - Oriented Object Programming (Programação Orientada a Objetos)
- PHP** - PHP Hypertext Processo (Processador de Hipertexto PHP)
- RIA** - Rich Internet Application (Aplicações de Internet Ricas)
- SBTVD-T** - Sistema Brasileiro de TV Digital - Terrestre
- URL** - Uniform Resource Locator (Localizador Padrão de Recursos)
- WYSIWYG** - What You See Is What You Get (O que você vê é o que você tem)
- XML** - eXtensible Markup Language (Linguagem de Marcação Extensível)

# Sumário

<b>Lista de Tabelas</b>	<b>6</b>
<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Siglas</b>	<b>8</b>
<b>1 INTRODUÇÃO</b>	<b>11</b>
1.1 Motivações . . . . .	12
1.2 Estrutura da Monografia . . . . .	12
<b>2 CONCEITOS E TECNOLOGIAS</b>	<b>14</b>
2.1 Sistema Brasileiro de TV Digital Terrestre e Ginga . . . . .	14
2.2 A Web como plataforma de aplicações . . . . .	14
2.3 XML . . . . .	16
2.4 NCL . . . . .	17
2.5 PHP . . . . .	18
2.6 JSON . . . . .	19
2.7 JQuery . . . . .	20
2.8 HTML5 . . . . .	21
2.8.1 Características . . . . .	21
2.8.2 Problemas . . . . .	23
2.9 Considerações . . . . .	24
<b>3 TRABALHOS ANTERIORES</b>	<b>25</b>
3.1 NCL Eclipse . . . . .	25

3.2	Composer . . . . .	26
3.3	TVision . . . . .	27
3.4	Comparativo . . . . .	28
<b>4</b>	<b>A FERRAMENTA CREATIVE</b>	<b>30</b>
4.1	Requisitos . . . . .	30
4.2	Detalhes de desenvolvimento . . . . .	31
4.2.1	Servidor . . . . .	31
4.2.2	Cliente . . . . .	32
4.2.3	Estrutura . . . . .	33
4.2.4	Templates . . . . .	33
4.3	Resumo . . . . .	34
<b>5</b>	<b>USO DA FERRAMENTA</b>	<b>35</b>
5.1	Tutorial . . . . .	35
5.2	Exemplos . . . . .	39
5.2.1	Aplicação “Qual a cidade?” . . . . .	39
5.2.2	Aplicação “Fotos de São Luís” . . . . .	41
5.3	Resumo . . . . .	42
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>43</b>
6.1	Trabalhos futuros . . . . .	43
6.2	Conclusões . . . . .	44

# 1 INTRODUÇÃO

Presente no cotidiano de grande parcela da população brasileira, a televisão ainda permanece como um dos maiores meios de comunicação, onde 95% dos domicílios possuem pelo menos um aparelho, que fica ligado por pelo menos cinco horas diárias em média (IBGE, 2009).

A transmissão digital de TV, que começou a operar a partir do final de 2007 no Brasil, é considerada a maior revolução tecnológica desde a sua invenção. Até 2014, a televisão analógica deverá ser extinta e a transmissão passará a ser exclusivamente digital.

A Alta Definição (mais conhecida através da sigla *HD*) é a característica mais associada à TV Digital, embora ela não seja a única novidade. Além de áudio e vídeo, a emissora pode transmitir, também, aplicativos diretamente para a casa do telespectador utilizando a largura de banda disponível. Esses aplicativos são responsáveis pela interatividade, que, além de agregar valor, tornam a experiência de assistir à televisão muito mais rica e personalizada.

Tais características podem ser exploradas tanto pela emissora quanto pelos anunciantes para enriquecer o conteúdo mostrado na tela de forma a manter presa a atenção do telespectador e com isso aumentar suas receitas. Esse novo modelo de negócios já começa a demandar uma série de novas ferramentas que possam auxiliar a produção desses aplicativos de maneira eficiente, até mesmo por usuários que não tenham conhecimento prévio de programação de computadores.

Este trabalho tem como principal objetivo desenvolver uma ferramenta de autoria para TV Digital cujo público alvo seja o usuário não-especialista. Seus objetivos específicos são:

- Fornecer um meio de criar aplicações interativas para TV Digital brasileira por quem não possui conhecimentos relacionados à programação de computadores;
- Utilizar o navegador como ambiente de criação;
- Prover uma interface intuitiva que forneça recursos de arrastar e soltar, e

controles gerados em tempo de execução;

- O trabalho final deve ter o código-fonte aberto e deverá ser disponibilizado em algum repositório público. Isso vai permitir a continuidade do mesmo.

## 1.1 Motivações

O padrão brasileiro de TV Digital define uma camada de software chamada Ginga, que é a plataforma responsável pela interatividade. O Ginga-NCL (Nested Context Language) é um dos subsistemas do Ginga utilizados na criação de aplicações interativas. Através da linguagem NCL são declaradas as mídias que serão exibidas, assim como as relações de sincronia temporal e espacial entre as mesmas. (Ginga-NCL, 2012).

Deve-se observar que o uso dessa linguagem é relativamente difícil por usuários não-programadores. Esse tipo de usuário será chamado daqui por diante de usuário final.

Recentemente, o governo impôs legalmente que, a partir do ano de 2013, 75% de todos os televisores fabricados no país terão de vir com o Ginga embarcado. E esse número vai aumentar até chegar à totalidade em 2015. O governo tem a pretensão de transformar a televisão em uma plataforma de inclusão digital (Terra, 2012).

Somado a isso, tem-se que, somente no ano de 2011, o mercado de propaganda televisiva movimentou cerca de 53 bilhões de reais (IBOPE, 2012). É um mercado que não pode ser ignorado, e cuja tendência é crescer cada vez mais.

Dada a demanda atual, e visando à futura, acredita-se que há espaço para a criação de novas ferramentas de autoria de aplicações interativas para TV Digital, tendo como foco o usuário final.

## 1.2 Estrutura da Monografia

Este trabalho está dividido nos seguintes capítulos:

- O Capítulo 2 expõe as tecnologias utilizadas ou relacionadas ao trabalho.
- O Capítulo 3 analisa trabalhos desenvolvidos anteriormente e que tenham relação com a monografia.
- O Capítulo 4 apresenta as necessidades identificadas (requisitos) para o desenvolvimento do trabalho e explica os detalhes de implementação.
- O Capítulo 5 demonstra um estudo de caso utilizando a ferramenta desenvolvida.
- O Capítulo 6 apresenta as principais conclusões, bem como propõe alguns tópicos a serem discutidos e implementados futuramente.

## 2 CONCEITOS E TECNOLOGIAS

Este capítulo apresenta as tecnologias e conceitos que são discutidos no decorrer deste trabalho.

### 2.1 Sistema Brasileiro de TV Digital Terrestre e Ginga

O padrão brasileiro de TV Digital, conhecido como SBTVD-T (Sistema Brasileiro de Televisão Digital - Terrestre), é baseado no padrão de transmissão japonês. O MPEG-4<sup>1</sup> foi adotado como *codec* oficial, pois oferece taxas de compressão maiores, permitindo que aplicativos com conteúdos mais ricos (maiores) possam ser desenvolvidos (Soares and Barbosa, 2012). Além do *codec*, o principal diferencial da TV Digital brasileira é a plataforma de desenvolvimento de interatividade.

Essa plataforma, conhecida como Ginga, é definida por norma da ABNT (Associação Brasileira de Normas Técnicas) e consiste na especificação de um padrão aberto de um *middleware*<sup>2</sup>. Esse *middleware* é dividido em dois subsistemas: o Ginga-NCL e o Ginga-J. O Ginga-NCL provê a estrutura de exibição de aplicações através de uma linguagem declarativa, baseada em XML (eXtensible Markup Language), chamada de de NCL, cujos conceitos básicos são explicados posteriormente.

Os aparelhos receptores, conhecidos como *Set-Top Boxes*, são capazes de executar aplicações interativas desde que contenham software que contemple essa especificação.

### 2.2 A Web como plataforma de aplicações

A Web, em seus primeiros dias, era uma coleção de textos que estavam ligados através de *hiperlinks*. A linguagem por trás desse hipertexto é o HTML

---

<sup>1</sup>MPEG-4 AAC para áudio e MPEG-4 AVC para vídeo

<sup>2</sup>Camada intermediária de software

(Hypertext Markup Language), cuja versão inicial (1991) tinha 22 *tags* (HTML, 2012).

O propósito inicial desta linguagem era simplesmente o de formatar o texto. Do navegador sempre foi a obrigação de interpretar e renderizar as páginas para visualização. Mesmo havendo uma especificação, a implementação oferecida por cada fabricante sempre foi individual, de modo que cada navegador dava suporte às *tags* e recursos de sua preferência. Dessa maneira, todo criador de páginas deveria ajustar o código HTML para que suas exibições fossem semelhantes, independente do navegador utilizado pelo usuário.

Percebeu-se que a Web era uma plataforma extremamente convidativa para o desenvolvimento de aplicações. A página passou a ser vista como um formulário que passava os dados para os servidores através de controles HTML simples (input, textarea, select) e exibiam os resultados deles obtidos. A maior vantagem dessa abordagem era que se tinha apenas um ponto de implantação e distribuição de aplicações. Não era mais necessário ter que instalar um sistema de computador em computador. A versão online seria sempre a mais atual. Outra vantagem que não podia ser negligenciada era a capacidade de a aplicação poder rodar em diversos Sistemas Operacionais sem maiores dificuldades. Não havia mais a necessidade de se compilar um sistema “n” vezes para rodar em diversas plataformas. Começou, dessa forma, uma onda de migração de aplicativos do desktop para a web. Nessa época, o termo “computação em nuvem” ainda nem existia.

Com o auxílio da linguagem Javascript, a Web tornou-se realmente dinâmica a ponto de permitir animações e manipulação de conteúdo da página. Podia-se, por exemplo, efetuar a validação dos formulários diretamente no cliente, para evitar que todo esse processamento fosse feito no servidor. Surgiu, então, uma tecnologia que recebeu o nome de AJAX (Asynchronous Javascript and XML), a qual permitia submeter dados ao servidor e tratar a resposta recebida sem que a página fosse recarregada. Apenas uma porção da mesma era atualizada em cada requisição. Dessa maneira, a aplicação se tornava mais rápida, dinâmica, e responsiva à medida em que menos dados precisavam trafegar entre o cliente e o servidor (Ajax, 2012).

A Web alcançou um patamar mais revolucionário ainda quando a criação do conteúdo passou para o próprio usuário (não-técnico). O foco passou, então, a ser a colaboração. Cunhou-se, assim, o termo WEB 2.0 (Web2.0, 2012).

A linguagem Javascript evoluiu para permitir, através de APIs (Application Programming Interface) cada vez mais poderosas, que o poder de processamento computacional do cliente fosse utilizado, ao invés de deixar toda a computação a cargo do servidor. Este ficava livre para a tarefa exclusiva de tratar os dados. A resposta dele recebida era, assim, tratada localmente.

Hoje, é fato que a Web consolidou-se como uma plataforma robusta de fornecimento de aplicações e serviços online, não só para computadores pessoais como também para dispositivos móveis, alavancados principalmente pelo aumento da velocidade de banda larga a preços mais baixos do que há alguns anos.

## 2.3 XML

Criada no fim da década de 90, a linguagem XML foi idealizada com a filosofia de descrever dados de maneira estruturada e de ser extensível, ou seja, um arquivo XML poderia ser escrito de forma a atender necessidades específicas. A Figura 2.1 mostra um exemplo de arquivo XML. Outro princípio que norteou o formato XML foi a característica semântica. Basta olhar para o conteúdo do arquivo para que se tenha uma ideia do que ele trata, através de suas *tags*. Como o conteúdo do arquivo é do formato texto, é legível tanto por seres humanos quanto por computadores. Figura como uma importante peça na interoperabilidade de sistemas, hoje em dia (Ray, 2003).

```
1 <tela>
2   <menu texto="Arquivo">
3     <submenu nome="Novo" icone="new.png"/>
4     <submenu nome="Abrir" icone="open.png"/>
5   </menu>
6   <componente nome="botao">Clique aqui</componente>
7   <pagina src="welcome.html" comprimento="100" largura="100" />
8 </tela>
```

Figura 2.1: Exemplo de arquivo XML

As regras para a criação de um XML bem formado são bastante simples:

- Todo documento XML deve ter um nó raiz;
- Toda *tag* de abertura tem que ter uma *tag* de fechamento;
- Toda *tag* pode ter como filho tanto texto quanto outras *tags*;

- Todo valor de atributo deve estar entre aspas.

Diversos protocolos e formatos de dados são baseados em XML. Como maior exemplo, temos o próprio NCL, explicado na seção seguinte.

Uma desvantagem da linguagem XML é que o arquivo pode se tornar verboso, já que a maior parte do conteúdo do mesmo estará nas próprias *tags* e não necessariamente na informação nelas contidas. Em época de dispositivos móveis, onde voltamos a nos preocupar com a quantidade de dados trafegados e poder de computação do cliente, pode não ser desejável. Entretanto, é uma solução ideal para arquivos de configuração ou dados que não serão transportados do servidor para o cliente.

## 2.4 NCL

A linguagem conhecida NCL foi proposta pela PUC-Rio, como uma alternativa de apresentação hipermídia interativa para ser utilizada na TV digital. Ela se baseia em XML, logo está sujeita às regras descritas na seção anterior.

Um documento NCL é um documento XML válido e bem-formatado. Uma aplicação básica NCL deve declarar, pelo menos:

- as regiões utilizadas na aplicação (*<region>*);
- os descritores de cada região, que especificam as propriedades das mesmas (*<descriptor>*);
- os conectores (exemplo: *<casualConector>*) que descrevem papéis (*roles*) desempenhados por cada elemento em condições e ações, quando associados a um evento (exemplo: *onBegin*);
- as mídias (*<media>*) que serão exibidas durante o tempo de vida da aplicação;
- os elos (*<link>*) que associam elementos a conectores, detalhando que papel cada um deles desempenhará;
- uma porta (*<port>*) de entrada para aplicação

É possível definir, ainda, através de *tags* NCL (Soares and Barbosa, 2012):

- relações de sincronia espaciais e temporais entre objetos de mídias, através de `<port>`, `<area>`, `<property>`, `<link>`, `<conector>`;
- exibição adaptativa de conteúdo, através de `<switch>` e `<descriptorSwitch>`;
- múltiplos dispositivos, através de `<regionBase>`;
- edição ao vivo, por meio de *nclEditingCommands* associados a descritores de eventos.

## 2.5 PHP

Essa linguagem foi criada por Rasmus Lerdof em 1995, como uma série de scripts pessoais para interpretar formulários e para produção de estatísticas do seu site. A linguagem evoluiu desde então, mas foi a partir da versão 3 que a linguagem alcançou notoriedade (PHP, 2012).

Um dos fatores que permitiram que ela tivesse esta trajetória é que sempre possuiu o código-fonte aberto e a sintaxe muito semelhante à linguagem C. Outro motivo é que pode ser facilmente integrada a um dos servidores Web mais conhecidos e utilizados no mundo: o Apache<sup>3</sup>.

Logo se formou uma grande comunidade ao redor da linguagem para adicionar funcionalidades não concebidas pelos mantenedores da linguagem.

Isso foi possível pois a versão 3 foi reescrita por Andi Gutman e Zeev Suraski, fundadores de uma companhia especializada em PHP – Zend.com, para permitir a extensão da linguagem através de módulos desenvolvidos em linguagem C. A partir daí surgiram várias extensões para os mais diversos fins: acesso a diversos bancos de dados, criptografia, funções de rede, socket e stream, imagem, acesso a arquivo, dentre outras. Algumas dessas extensões foram embutidas oficialmente na linguagem de acordo com a qualidade das mesmas e as funcionalidades proporcionadas.

---

<sup>3</sup>Disponível em [http://projects.apache.org/projects/http\\_server.html](http://projects.apache.org/projects/http_server.html)

A versão 4 do PHP marcou a introdução da Programação Orientação a Objeto (POO), mesmo que bastante modesta – havia apenas a ideia de classes e todos os atributos eram públicos. Esse conceito só foi amadurecido a partir da versão 5 (como mostrado na Figura 2.2), quando a linguagem ofereceu meio de descrever os tipos de acesso dos atributos (`public`, `protected` e `private`), classes abstratas e interfaces, atributos e métodos estáticos e, mais recentemente, namespaces (Loudon, 2010). Atualmente, encontra-se na versão 5.4.

```
<?php
include 'IClass.php';

class Class1 extends BaseClass implements IClass
{
    private $attribute;

    public function __construct (BaseClass $Object)
    {
        $this->attribute = $Object->getAttribute();
    }
?>
```

Figura 2.2: Orientação a Objetos em PHP

É interessante notar que embora ela dê amplo suporte à POO, ela é uma linguagem que ainda permite a programação procedural. Fica a critério do programador definir o melhor meio de usá-la.

Essas características tornaram o PHP o que é hoje: uma das linguagens de processamento server-side mais utilizadas no mundo (LangPop, 2012).

## 2.6 JSON

O JSON<sup>4</sup> é um formato de dados bastante leve utilizado para a troca de informações. Utiliza formato texto e conta com duas estruturas básicas para descrever os dados:

- Uma coleção de pares "nome-valor". Essa estrutura recebe vários nomes diferentes de acordo com a linguagem a ser utilizada. *Array* associativo, *hashtable*, *dictionary*, e outros.
- Uma lista ordenada de valores. Geralmente expressa por *arrays*, *vectors* ou *lists*.

<sup>4</sup>Especificação disponível em <http://json.org>

Tem a vantagem de ser mais compacta do que XML, e igualmente hierárquico. Por outro lado, perde-se a característica semântica.

É considerada a melhor opção para transferência de dados via AJAX e é a língua *de facto* em muitos frameworks, já que é o formato interno que o Javascript utiliza para armazenar dados, o que se traduz, também, em velocidade para interpretá-los.

## 2.7 JQuery

Fabricantes diferentes dão suporte a *tags*, estilos em cascata próprios, métodos e classes Javascript diferentes. Mesmo navegadores iguais dispõem de elementos diferentes, dependendo da versão. Por exemplo: escrever um código javascript puro que manipula o DOM (Document Object Model), que rode em todos os navegadores, é uma tarefa trabalhosa e bastante sujeita a erros.

Por causa disso, em 2006, John Resig, cansado de escrever várias versões de código para vários navegadores diferentes, resolveu desenvolver uma biblioteca que ficasse uma camada acima do javascript nativo e tratasse da diferença existente entre os vários navegadores. Ele chamou essa biblioteca de JQuery. Atualmente se encontra na versão 1.7.2<sup>5</sup>.

Ele usou o termo “query” porque se baseou na filosofia do CSS (Cascade Stylesheets), o qual faz uso extensivo de seletores para navegar através do DOM até chegar ao elemento desejado. Essa operação de busca é conhecida por *DOM Traversal*. Após selecionar os elementos desejados, é possível executar operações sobre os mesmos.

A variável global \$ é um *alias* para o objeto JQuery e é ela que permite selecionar elementos. Por exemplo: o código \$('p') seria suficiente para selecionar todos os elementos cuja *tag* seja <p>. Existem ainda seletores de id (#) e de classe (.) e ainda pseudo-seletores (como *:first* e *:checked*) que permitem efetuar seleções com facilidade. Com a seleção em mãos, pode-se aplicar animações, atribuir ou obter propriedades ou dados, submeter informações da mesma ao servidor, via ajax, e associá-la a eventos (tais como clique ou duplo-clique). O código vai rodar sem

---

<sup>5</sup>Disponível em <http://www.jquery.com>

modificações nos diversos navegadores (Castledine and Sharkie, 2012).

Outras características importantes do Jquery são o tamanho reduzido – apenas 32Kb, a versão de produção – e o fato de ser extensível. Existem centenas de *plugins* que executam das tarefas mais simples às mais complexas. Componentes inteiros podem ser construídos com uma simples chamada a um método, por exemplo. O Jquery fornece, ainda, diversos métodos utilitários que não atuam especificamente sobre seleções e permite, da mesma forma, que possam ser estendidos para adicionar novas funcionalidades.

O grupo Jquery mantém um outro projeto paralelo, o Jquery UI, que disponibiliza diversos controles, métodos e efeitos. Esses itens não fazem parte do núcleo original e foram construídos utilizando a estrutura de *plugins* (Sarrion, 2012). Inclusive, pode-se montar uma versão personalizada dessa biblioteca diretamente no sítio do projeto, contendo apenas as classes de interesse, diminuindo-se, dessa forma, tamanho final de *download* da biblioteca. Encontra-se, atualmente, na versão 1.8.20<sup>6</sup>.

## 2.8 HTML5

Em 2008, começaram a ser discutidas as propostas iniciais para a nova especificação da linguagem HTML. Considerando que ela ainda é um rascunho, já se tem mais de uma década que a versão 4.01 permanece.

As versões mais recentes dos principais navegadores já contemplam alguns itens da especificação do HTML5, enquanto outros itens ainda não alcançaram unanimidade.

### 2.8.1 Características

O HTML versão 5 foi (e continua sendo) moldada para alavancar a característica da Internet como plataforma de aplicações e serviços. Essa versão oferece os seguintes recursos (Kessin, 2012):

- Armazenamento local de dados: até 5MB de dados armazenados como

---

<sup>6</sup>Disponível em <http://www.jqueryui.com>

estruturas do tipo chave-valor;

- Bancos de dados no navegador: implementados em Javascript e que indexam objetos e documentos no formato JSON;
- Geolocalização: expõe métodos Javascript para obter a localização do dispositivo cliente;
- Arquivos: embora o navegador não consiga acessar o sistema de arquivos livremente, por questões óbvias de segurança, ele passa a ter a permissão de acesso a um arquivo especificado diretamente pelo usuário;
- Modo *offline*: permite especificar como efetuar cache para uso posterior.
- *Web Workers*: implementação de *threads* no navegador;
- *Web Sockets*: permite a comunicação bidirecional *full-duplex* TCP e não possui todo o *overhead* de uma comunicação HTTP tradicional (define os esquemas de URLs *ws://* e *wss://* para conexões descriptadas e encriptadas, respectivamente);
- Novas tags: a especificação criou novos elementos com os mais variados propósitos.
  - Semânticas: ao invés de utilizar `<div>` e `<span>` para quase tudo, o desenvolvedor pode dar mais significado ao conteúdo através das *tags* `<nav>`, `<menu>`, `<session>`, `<header>`, `<footer>`;
  - Multimídia: novas marcações permitem adicionar áudio e vídeo às páginas; alguns navegadores implementam uma tela (`<canvas>`) na qual pode-se desenhar através de código Javascript;
  - Controles: os formulários podem ser enriquecidos por meio de novos componentes do tipo `<input>` que reconhecem *e-mail*, telefone, data, cores e números, livrando o desenvolvedor de ter de reescrevê-los a cada aplicação. Alguns navegadores implementam, ainda, a *tag* `<progressbar>`, que servem para indicar a porcentagem de conclusão de uma determinada tarefa.

Mesmo não fazendo parte da especificação HTML5, é importante comentar que o CSS3 tem papel fundamental nessa nova forma de se pensar aplicações. Algumas características em uso hoje em dia, como por exemplo webfonts e layouts responsivos, são possíveis graças a novos elementos que foram trazidos juntamente com esta versão.

## 2.8.2 Problemas

Apesar da promessa de aplicações melhores e mais funcionais, não se deve esquecer de que mesmo estando disponível há quase 5 anos, a especificação ainda está em estado de rascunho (*Working Draft*). De lá para cá, algumas coisas já mudaram. Por exemplo, a especificação Web SQL caiu em desuso, sendo substituída por outra chamada IndexedDB (Kessin, 2012). Outras coisas ainda podem estar sujeitas à mudanças.

Como explicado anteriormente, cada fabricante implementa as *tags* de sua preferência, APIs que desejam e escolhem que porções da especificação serão contempladas.

A especificação multimídia é um exemplo bem característico dessa “guerra” entre fabricantes e ela é melhor exemplificada na tentativa de padronização dos *codecs* necessários para rodar áudio e vídeo diretamente no navegador. A fundação Mozilla, por exemplo, recusa-se a disponibilizar o *codec* MP4 alegando que os *royalties* anuais são excessivos. O Google, por sua vez, suporta o mp4 no seu navegador Chrome, mas aproveitou a situação para promover o seu *codec* Vorbis Ogg/WebM na tentativa de torná-lo o próximo padrão de vídeo na Internet, e desbancar assim o império do MP4/H264/AAC, famoso principalmente por causa do Youtube (Pfeiffer, 2010). A tentativa, parece, tem surtido efeito já que a maioria dos principais navegadores conseguem, nativamente ou por meio de *plugins* de terceiros, reproduzir vídeos no formato Vorbis. Isso traz o problema de se ter que disponibilizar duas versões de um determinado vídeo, para que todos os clientes consigam tocá-lo independente do navegador.

Para sanar, em parte, essa dificuldade, existem bibliotecas que emulam o funcionamento de APIs em navegadores que não as implementam nativamente. Existe

um projeto de nome HTML5 Cross-Browsers Polyfills<sup>7</sup> cujo objetivo é catalogá-las. Ele é mantido pelo grupo criador da biblioteca *Modernizr* que serve para descobrir quais funcionalidades determinado navegador suporta (ou não) e permite escolher estilos ou, ainda, implementar controles diferentes, dependendo da existência destas funcionalidades. “Aproveitar-se das novas funcionalidades do HTML5 e CSS3 pode significar sacrificar o controle sobre a experiência em navegadores antigos” (Modernizr, 2012), caso uma dessas alternativas não seja utilizada.

## 2.9 Considerações

As aplicações Web tem evoluído e as tecnologias descritas neste capítulo acompanham essa evolução para permitir o desenvolvimento de aplicações cada vez mais robustas. A linguagem HTML5 é um bom exemplo de evolução para melhor. Porém, enquanto ela não for uma especificação final, ao mesmo tempo em que fornece elementos úteis e poderosos, será uma linguagem trabalhosa de se utilizar, visto que nem todos os navegadores a interpretam da mesma maneira.

---

<sup>7</sup>Disponível em <https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills>

## 3 TRABALHOS ANTERIORES

Este capítulo descreve algumas ferramentas desenvolvidas, dentre várias, que tenham o intuito de facilitar a criação de aplicações interativas para TV Digital.

### 3.1 NCL Eclipse

Desenvolvido no laboratório LAWS (Laboratory of Advanced Web Systems) da Universidade Federal do Maranhão em parceria com a PUC-RIO, o NCL Eclipse (atualmente na versão 1.6) é, atualmente, a ferramenta mais utilizada para criação de aplicações Ginga-NCL. Ela nasceu com o intuito de tornar o trabalho de autores de apresentações interativas mais produtiva e menos propensa a erros.

O NCL Eclipse é um *plugin* de código-fonte aberto para o Ambiente de Desenvolvimento Integrado Eclipse, muito conhecido e robusto. Através do conceito de pontos de extensão deste ambiente, diversos *plugins* podem se acoplar ao editor para fornecer as mais diversas funcionalidades.

Essa ferramenta tem amadurecido bastante em suas últimas versões e oferece suporte aos idiomas português, inglês e espanhol, visualização de leiaute, pré-visualização de mídias e imagens (conforme Figura 3.1), sugestão de código, marcação de erros e sugestões para correção dos mesmos (Azevedo et al., 2009).

Apesar de ser voltada para o usuário com perfil de programador, acredita-se que a funcionalidade de pré-visualização pode ser utilizada em ferramentas voltadas para outros perfis de usuários.

Informações adicionais a respeito da ferramenta podem ser obtidas na página oficial da mesma<sup>1</sup>.

---

<sup>1</sup>Disponível em <http://laws.deinf.ufma.br/ncleclipse>

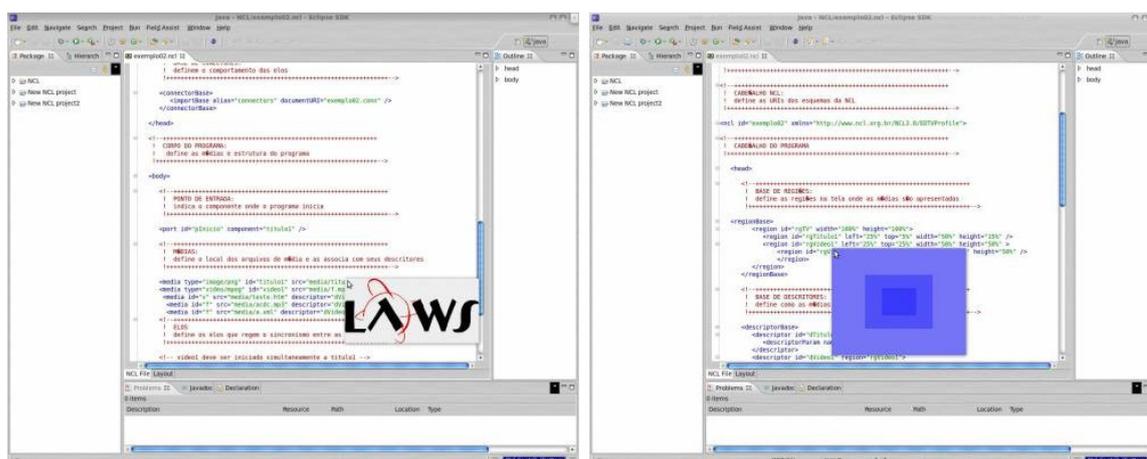


Figura 3.1: NCL Eclipse - visualização de Mídia e de Região

## 3.2 Composer

A ferramenta Composer é um ambiente de desenvolvimento de documentos NCL implementado pelo laboratório Telemídia da PUC-Rio<sup>2</sup>. Seu intuito primário é oferecer recursos avançados a partir de interface gráfica e elementos visuais, de acordo com a visão desejada: estrutural, temporal, leiaute, de código, etc. (Guimarães, 2007).

Como a versão anterior sofria de alguns problemas de performance e usabilidade, ela foi reformulada e reescrita do zero, para que esses problemas pudessem ser sanados. Recentemente, foi lançada a versão 0.1.0 beta. Esta nova versão utiliza a linguagem C++ com biblioteca Qt, de forma a melhorar a performance, mas mantendo a portabilidade, e oferece assim uma melhor experiência no desenvolvimento de aplicações NCL. Por ser baseada em uma estrutura orientada à *plugins*, pode ser estendida de acordo com a necessidade.

Mesmo utilizando conceitos exclusivos da linguagem NCL como conectores, elos e áreas, por exemplo, a ferramenta é de fácil manuseio por usuários quem não detêm vastos conhecimentos de programação e isso pode ser considerado um avanço no que diz respeito à ferramentas de autoria NCL. A Figura 3.2 exhibe diversas visualizações simultâneas.

<sup>2</sup>Disponível em <http://composer.telemidia.puc-rio.br>

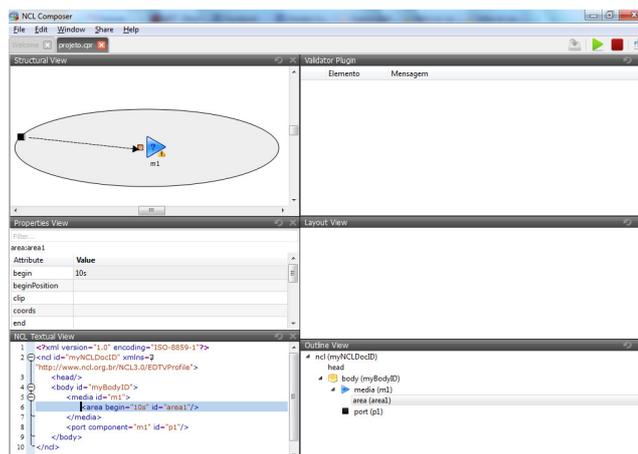


Figura 3.2: Novo Composer mostrando diversas visualizações simultaneamente

### 3.3 TVision

Desenvolvido em 2009 como trabalho final de graduação na Universidade Federal de Pernambuco, o TVision (Oliveira, 2009) foi idealizado como uma ferramenta gráfica cujo objetivo é auxiliar o desenvolvimento de aplicações NCL. Ela ainda se encontra disponível e pode ser acessada até hoje. A Figura 3.3 mostra a página inicial da ferramenta.

Com foco na tecnologia Flash<sup>3</sup> e utilizando PHP no lado do servidor, obteve-se uma aplicação RIA (Rich Internet Application) que implementa controles básicos como caixa de textos, imagens e botões, utilizados como base para construir aplicações simples, e fornece um meio de alterar as propriedades dos mesmos.

Segundo o seu autor, o projeto logrou um certo êxito à medida em que conseguiu construir uma aplicação simples como estudo de caso, sem utilizar a linguagem NCL para isso, mesmo que a interface com o usuário não fosse a mais amigável possível.

<sup>3</sup>Disponível em <http://www.adobe.com/br/products/flash.html>

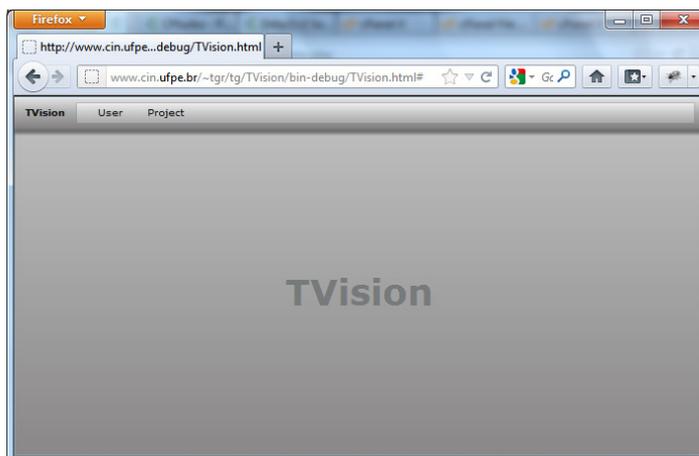


Figura 3.3: Ferramenta TVision

Durante a pesquisa deste trabalho, foi realizada uma tentativa de sua utilização, que não se mostrou satisfatória. Ela não funcionou como esperado, ao impedir, por exemplo, de posicionar o controle texto (conforme exibido na Figura 3.4) e não permitindo a criação de controles “imagem” e “botão”. Supõe-se que isso se deve ao fato de a aplicação ter sido desenvolvida para uma versão anterior do Flash.

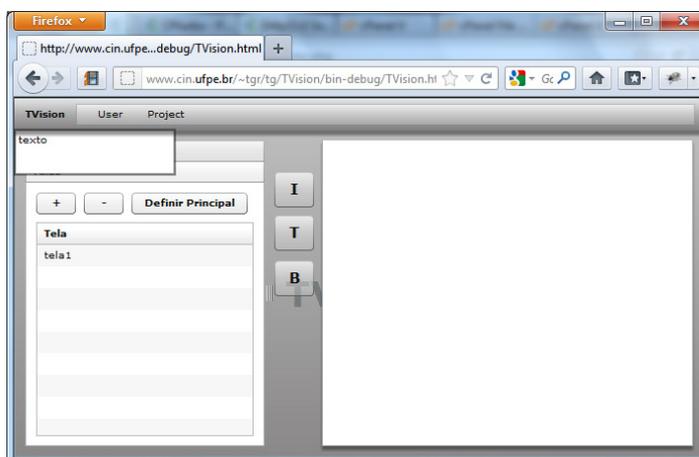


Figura 3.4: Tvision - Problemas na utilização

### 3.4 Comparativo

A tabela 3.1 abaixo mostra uma análise comparativa das ferramentas analisadas neste capítulo:

Tabela 3.1: Comparativo das ferramentas descritas

Ferramentas	Pontos fortes	Pontos fracos
NCL Eclipse	<ul style="list-style-type: none"> <li>• Integrado ao Eclipse</li> <li>• Testes integrados com máquina virtual Ginga;</li> <li>• Visualização de regiões e de mídias;</li> </ul>	<ul style="list-style-type: none"> <li>• Conhecimento de NCL obrigatório.</li> </ul>
Composer	<ul style="list-style-type: none"> <li>• Conhecimento de NCL não obrigatório, embora sugerido;</li> <li>• Interface Gráfica;</li> <li>• <i>Plugins</i></li> </ul>	<ul style="list-style-type: none"> <li>• Sem abstração WYSIWYG.</li> <li>• Visualização de propriedades sem controles especializados.</li> </ul>
TVision	<ul style="list-style-type: none"> <li>• Nenhum Conhecimento de NCL exigido;</li> <li>• Roda direto no navegador.</li> </ul>	<ul style="list-style-type: none"> <li>• Interface com usuário deixa a desejar;</li> <li>• Descontinuada.</li> </ul>

Baseado na análise efetuada nesse capítulo, acredita-se que a necessidade de ferramentas de autoria que possibilite o usuário final criar aplicações para TV Digital, sem conhecimento de NCL, exista.

## 4 A FERRAMENTA CREATIVE

Este capítulo trata de aspectos específicos da implementação da ferramenta creaTiVe, proposta deste trabalho, para a autoria NCL.

### 4.1 Requisitos

Esta seção enumera as necessidades que se deseja suprir com a ferramenta. A lista de requisitos foi criada após pesquisa e análise dos trabalhos citadas na seção anterior e também com base em aplicativos selecionados no Clube NCL (Clube-NCL, 2012). Ela é citada abaixo:

- RF01 – O sistema deverá permitir a criação de projetos.
- RF02 – O sistema deverá permitir a abertura de projetos para edição.
- RF03 – O sistema possibilitará que o usuário salve o projeto, para continuar trabalhando nele no futuro.
- RF04 – O sistema deverá utilizar sistema de *templates* já prontos, mas terá de prover customização dos mesmos.
- RF05 – O sistema deverá dispor de elementos gráficos para que o usuário manipule as regiões, posicionando-as e redimensionando-as de acordo com a necessidade.
- RF06 – O sistema deverá fornecer um meio de arrastar e soltar elementos sobre outros elementos, alterando assim as devidas propriedades.
- RF07 – O sistema deverá proporcionar a mudança do idioma de exibição com facilidade.
- RF08 – O sistema deverá possibilitar que o usuário efetue upload de arquivos, os quais poderão ser relacionados às mídias.

- RF09 – O sistema terá de permitir a alteração das propriedades de seus objetos visuais.
- RF10 – O sistema deverá expor uma opção para exportar o projeto final NCL, de preferência em formato compactado.
- RNF01 – O projeto deverá ter uma licença de uso permissiva.
- RNF02 – O sistema terá seu código disponibilizado publicamente em um repositório gratuito.
- RNF03 – A aplicação final deverá rodar no maior número de navegadores possível.

## 4.2 Detalhes de desenvolvimento

A ferramenta creaTiVe foi desenvolvido em plataforma Windows, utilizando o editor NetBeans (versão 7.0.1) para programação de arquivos PHP, Javascript e edição de arquivos XML. Testes efetuados em ambientes Linux mostraram-se satisfatórios.

### 4.2.1 Servidor

Para o servidor rodando a linguagem PHP, utilizou-se o aplicativo WampServer (versão 2.0c), que instala os servidores Apache e PHP em Sistemas Operacionais Windows (juntamente com o MySQL, embora não tenha sido utilizado no projeto).

A linguagem PHP é utilizada para todos os eventos que ocorrem do lado do servidor. São eles:

- Leitura dos arquivos de *template*, e dos arquivos de tradução de idioma;
- Para abrir, salvar e exportar um projeto;
- Para permitir o *upload* de arquivo, adicionando-o ao projeto ativo.

No início da aplicação, lê-se as respectivas pastas de *templates* e de tradução. Os detalhes de *templates* serão explicados posteriormente nesse capítulo. Os arquivos de tradução são arquivos XML de termos no formato chave-valor, conforme demonstrado na Figura 4.1:

```
<Translation>
  <Key name="PROPERTIES" value="Propriedades"/>
  <Key name="VIEW" value="Visualizar"/>
  <Key name="TYPE" value="Tipo"/>
  <Key name="PROJECT" value="Projeto"/>
```

Figura 4.1: Trecho do arquivo de tradução pt-br

Quando da criação de um projeto, é criada uma pasta no servidor para conter os arquivos do mesmo. Também é criado um arquivo *project.xml*, que contém as informações relativas ao projeto. Salvando-se ou acrescentando-se arquivos a um projeto, seu arquivo XML é modificado para refletir as mudanças.

## 4.2.2 Cliente

Foi eleita a biblioteca JQuery para se implementar as características dinâmicas que são executadas no cliente, principalmente manipulação de DOM e AJAX. Dessa forma, espera-se suportar o maior número possível de navegadores.

A ferramenta faz uso intensivo de caixas de diálogos para abrir projetos, *upload* de arquivos, visualização de mídia, e para o efeito de arrastar e soltar. Utilizou-se a biblioteca JQuery UI, com o tema *redmond*, para implementar essas funcionalidades.

Foi utilizado um plugin jQuery chamado jsRender (JSRender, 2012), cujo objetivo é formatar um objeto JSON de acordo com um *template* especificado. Dessa forma, houve uma melhor separação entre a visualização e os dados propriamente ditos. Se não fosse assim, seria preciso construir, em tempo de execução, uma *string* com o código HTML e adicioná-la ao DOM.

Outra biblioteca utilizada do lado cliente tem o nome de Video-js (VideoJS, 2012). Através dela, pode-se utilizar a tag `<video>` do HTML5, independente do suporte do navegador. Se ela for suportada, então a *tag* é utilizada. Em caso contrário, ou se o formato do vídeo não for aceito, (como por exemplo: arquivo mp4 no navegador

Mozilla Firefox), então a *tag* é substituída por um objeto *Flash* que simula a aparência e funcionalidade de um *player* de vídeo HTML5 nativo.

### 4.2.3 Estrutura

A estrutura dos arquivos da ferramenta creaTiVe é muito simples e para implantá-la basta copiar a sua pasta para um diretório qualquer do *DocumentRoot* do Apache (no WampServer, por padrão, essa pasta se encontra em */wamp/www*).

O arquivo *.htaccess*, na pasta raiz da ferramenta, configura o PHP para permitir upload de arquivos grandes, já que a premissa do projeto antecipa que serão utilizados arquivos de vídeo. Utilizou-se como padrão, o valor limite de 100MB, pois julgou-se ser suficiente. Basta editar o arquivo *.htaccess* para permitir valores maiores. Isso implicará na necessidade de mais disponibilidade de memória e espaço em disco no servidor utilizado.

Na pasta *classes*, encontram-se as classes PHP, divididas entre classes da própria ferramenta (exemplo: gerenciamento de projetos) e classes que manipulam os *templates*. Na pasta *projects*, ficam os projetos já criados. A pasta *service* abriga as ações que podem ser executadas. Essas ações são codificadas como scripts php que tem, em média, 50 linhas de código. As demais pastas são auto-explicativas: *images*, *css*, *translations*, etc.

### 4.2.4 Templates

O formato de *template* utilizado na ferramenta creaTiVe é composto por dois arquivos XML.

Os templates propriamente ditos são arquivos NCL válidos, segundo as seguintes restrições:

- As regiões não deve ser aninhadas.
- Todos os valores de dimensões das regiões devem ser especificados em porcentagem.

O autor do *template* deve criar, também, um arquivo TIIF (Template Intermediate Information File). Esse formato foi criado especificamente para a ferramenta. Isso foi necessário para não alterar o arquivo de template NCL, já que isso poderia torná-lo inválido. Dessa maneira, tem-se dois arquivos XML válidos que podem ser lidos através da extensão PHP chamada *SimpleXML*.

O arquivo TIIF descreve as mídias que estão disponíveis. Elas podem ser tanto obrigatórias como opcionais. Fica a cargo do autor do *template* documentar cada uma delas. A Figura 4.2 mostra um exemplo de arquivo TIIF.

```
<Template id="vivamais">
  <media id="video" region="rgVideo" type="video" zindex="2"/>
  <media id="icone" region="rgIcane" type="image" zindex="4"/>
  <media id="fundo" region="rgBackground" type="image" zindex="1"/>
  <media id="vermelho" region="rgVermelho" type="image" zindex="3"/>
  <media id="amarelo" region="rgAmarelo" type="image" zindex="3"/>
  <media id="verde" region="rgVerde" type="image" zindex="3"/>
  <media id="azul" region="rgAzul" type="image" zindex="3"/>
  <media id="instrucoes" region="rgTextos" type="image" zindex="3"/>
  <media id="resultadovermelho" region="rgTextos" type="image" zindex="3"/>
  <media id="resultadoamarelo" region="rgTextos" type="image" zindex="3"/>
  <media id="resultadoverde" region="rgTextos" type="image" zindex="3"/>
  <media id="resultadoazul" region="rgTextos" type="image" zindex="3"/>
</Template>
```

Figura 4.2: Exemplo de arquivo TIIF extraído do template “vivamais”

A leitura do arquivo de *template* juntamente com o arquivo TIIF gera o arquivo de projeto (*project.xml*), explicado anteriormente.

## 4.3 Resumo

Este capítulo especifica os requisitos levantados para o projeto e dá detalhes de como essas funcionalidades foram implementadas. O próximo capítulo mostra como utilizar a ferramenta para criar aplicações interativas baseadas em *templates*.

## 5 USO DA FERRAMENTA

Este capítulo demonstra a utilização da ferramenta creaTiVe e mostra exemplos de aplicações finais geradas na mesma.

A Figura 5.1 mostra a tela da aplicação.

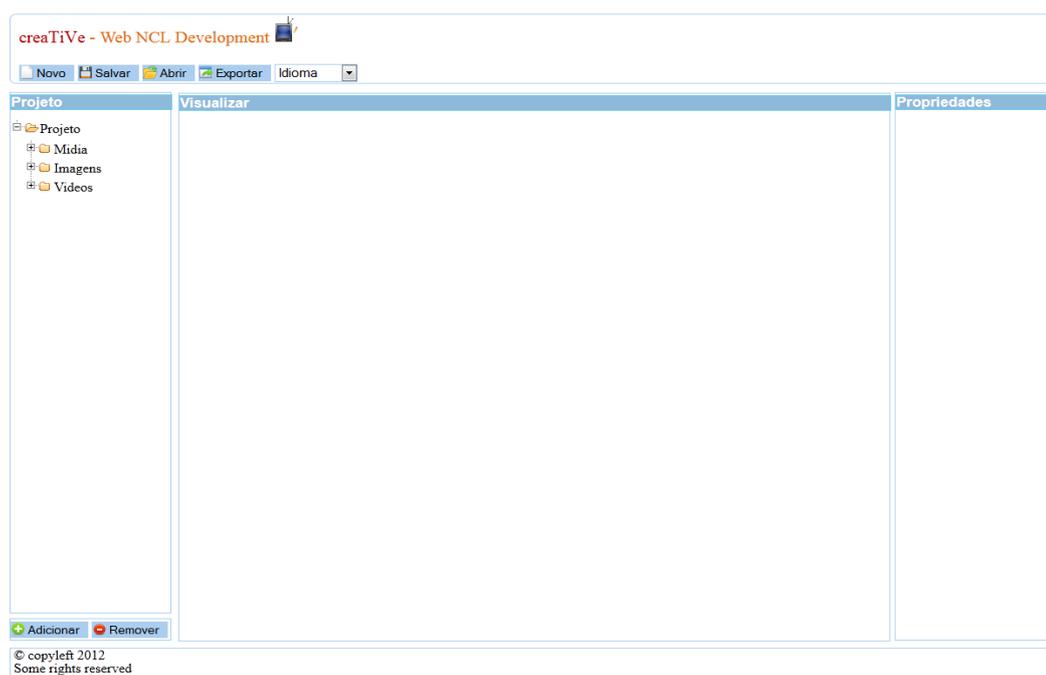


Figura 5.1: Tela inicial da ferramenta

### 5.1 Tutorial

A barra de ferramentas dá acesso aos recursos disponíveis para o autor de aplicações, como exibido na Figura 5.2.



Figura 5.2: Barra de ferramentas

Conforme dito anteriormente, um *template* é uma aplicação NCL completa. Ele serve como modelo no qual novas aplicações podem se basear. Neste capítulo foram utilizados dois. A Figura 5.3 mostra as aplicações que originaram os mesmos. A aplicação da esquerda originou o *template* chamado “vivamais”. A da direita, “portaretrato”. Os detalhes destes *templates* são especificados na próxima seção.



Figura 5.3: Aplicações NCL originais

Ao iniciar um novo projeto (Figura 5.4), é possível escolher dentre os *templates* disponíveis na pasta “*templates*” no servidor. Deve-se dar um nome único ao projeto. Caso já exista algum com o nome inserido, será necessário escolher um novo nome.

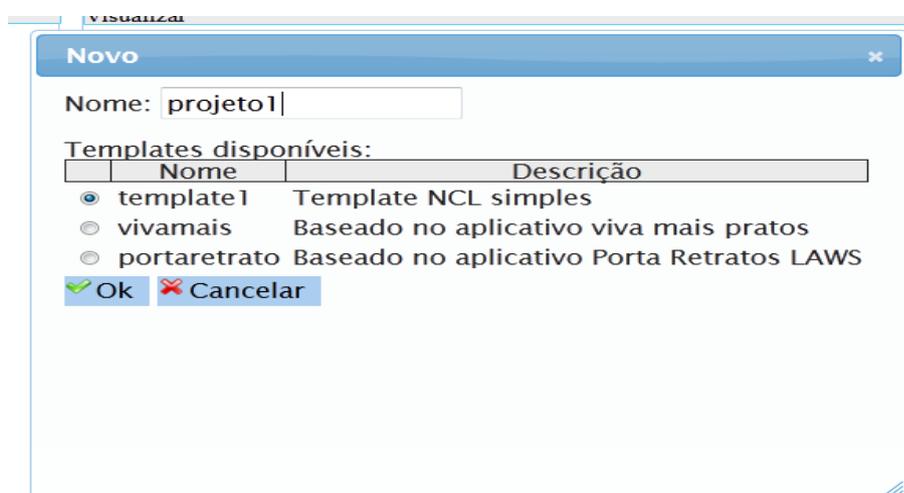


Figura 5.4: Criando um novo projeto

Pode-se, então, efetuar o *upload* (Figura 5.5) de arquivos do tipo imagem e vídeo (no formato mp4), para que sejam associados às mídias disponíveis no *template*, através da operação de arrastar e soltar. Os arquivos são adicionados à árvore do projeto, de acordo com o seu tipo. Para que uma associação seja possível, é necessário se certificar se o tipo da mídia é igual ao tipo de arquivo desejado, ou seja, se os seu ícones são iguais.



Figura 5.5: Adicionando arquivos

Ao repousar o mouse sobre um arquivo, o nome original do arquivo é exibido (Figura 5.6).

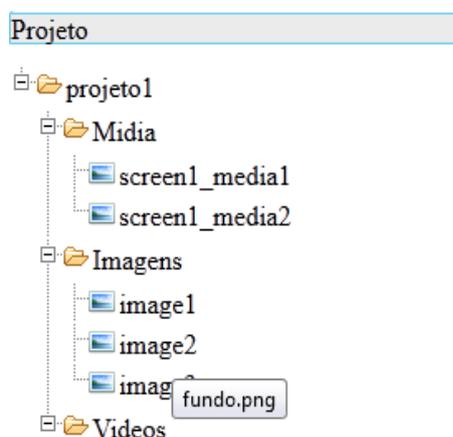


Figura 5.6: Árvore de Projeto

Clicando-se em um arquivo, tem-se uma visualização do mesmo. A janela de visualização vai exibir o conteúdo de acordo com o tipo. Se for uma imagem, ela será exibida. Caso seja um vídeo, exibe-se um tocador de vídeo, que pode ser HTML5 nativo, caso o navegador dê suporte a arquivos MP4 ou um player Flash, em caso contrário (Figuras 5.7 e 5.8).



Figura 5.7: Visualizador de Imagem



Figura 5.8: Tocador de vídeo

É importante ressaltar que durante o desenvolvimento, o tocador de mp4, independente de ser a versão nativa ou Flash, pode acabar reproduzindo apenas o áudio, sem exibir o vídeo em si. Isso é ocasionado, principalmente, por problemas de compatibilidade de *codecs*. Entretanto, ao serem executados no Set-top Box Virtual do Ginga, os vídeos rodam normalmente.

O autor pode, também, posicionar as regiões conforme desejado, bem como redimensioná-las e pode, ainda, arrastar uma mídia e soltá-la sobre uma região, para que nela seja exibida. Assim, o usuário tem mais liberdade sobre onde exibir a mídia.

Ao clicar sobre uma mídia ou sobre uma região, tem-se informações atualizadas das mesmas, conforme pode ser visto na Figura 5.9.

Após todas as configurações necessárias, basta exportar o projeto, que será disponibilizado para *download* por meio de arquivo compactado em formato zip (project.zip). Ele contém todos os arquivos necessários para rodar a aplicação gerada

screen1_media2		region1	
Id:	screen1_media2	Top:	49.64%
Type:	image	Left:	29.33%
Region:	region2	Width:	40%
Source:	image2	Height:	40%
		zIndex:	2

Figura 5.9: Propriedades de mídia e região

no Set-top Box Virtual do Ginga.

## 5.2 Exemplos

Essa seção demonstra dois exemplos de aplicações desenvolvidas com a ferramenta creaTiVe.

### 5.2.1 Aplicação “Qual a cidade?”

Objetivo: exibir um vídeo e permitir a interação com o usuário através de pergunta de múltipla escolha.

Nessa aplicação, a interação se dá da seguinte forma: o usuário, através das teclas coloridas do controle remoto, vai tentar adivinhar qual a cidade exibida no vídeo. Caso, ele acerte, receberá uma mensagem de sucesso. Caso contrário, será exibido um aviso informando que ele não acertou. O vídeo segue tocando até o final, independente da interação com o usuário.

Utilizou-se o *template* “vivamais”. A aplicação foi desenvolvida pelo Laboratório Telemídia da PUC-Rio e tem licença *Creative Commons*. Encontra-se disponível no Clube NCL. Ao ser convertida para *template*, tornaram-se opcionais o ícone de interatividade e a imagem de fundo. Neste exemplo, apenas o ícone opcional foi deixado intacto. Todos os demais itens foram personalizados.

As mídias desse *template* têm identificadores bastante intuitivos, conforme a Figura 5.10. Essa prática, aliás, é sugerida aos criadores de *templates*. É bem melhor identificar uma mídia como “resultadovermelho” do que “resultado3”, por exemplo. Utilizou-se uma imagem do tipo *png* para cada cidade. Foi escolhida a

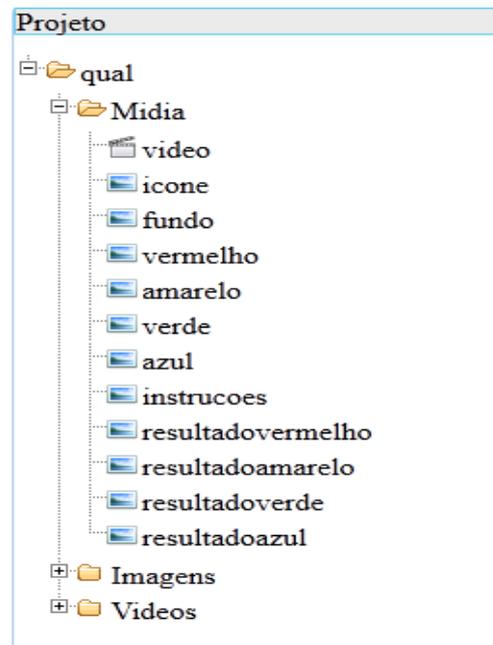


Figura 5.10: Mídias do template “vivamais”

cor vermelha para a cidade correta: São Luís. As demais cidades utilizadas foram: Fortaleza, Belém e Teresina. Foi atribuída a imagem “correta” ao resultado vermelho. As demais mídias de resultados receberam a imagem “incorreta”. A Figura 5.11 mostra a aplicação.



Figura 5.11: Aplicação “Qual a cidade?” exibindo a resposta correta

## 5.2.2 Aplicação “Fotos de São Luís”

Objetivo: exibir um vídeo e permitir a interação com o usuário através de álbum de fotos.

Nessa aplicação, a interação se dá da seguinte forma: enquanto é exibido um vídeo da cidade de São Luís, o usuário tem a opção de exibir um álbum com diversas fotos da cidade. Ao terminar, ele pode voltar e ver o final do vídeo.

Foi utilizado o *template* “portaretrato”. A aplicação foi desenvolvida por Thiago Alencar, Almir Filho e José Rios, alunos do laboratório LAWS, da UFMA, tem licença *Creative Commons* e também está publicada no Clube NCL. Ao ser convertida para *template*, tornaram-se opcionais todos os ícones de botões, bem como a imagem de fundo. Neste exemplo, todos os itens opcionais foram mantidos. Apenas as fotos do álbum foram personalizadas.

Na versão atual do *template*, as mídias não têm identificadores muito intuitivos, por isso, uma rápida explicação é necessária. Aquelas cujos identificadores começam com “Bt” referenciam botões; as que começam com “foto” fazem referência às miniaturas; as que iniciam apenas com “ft” dizem respeito à foto central exibida quando alguma miniatura é selecionada pelo usuário. Foi atribuída a mesma imagem para as mídias “foto” e “ft” correspondentes para que a aplicação funcione corretamente, mostrando a respectiva imagem central. A Figura 5.12 mostra a aplicação.



Figura 5.12: Aplicação “Álbum de São Luís” sendo exibida

### 5.3 Resumo

Este capítulo demonstrou os recursos disponíveis na ferramenta creaTiVe e os passos necessários para o autor criar um aplicativo do início até o final. Foram descritas também dois exemplos de aplicações creaTiVe baseadas em *templates* diferentes. O capítulo a seguir analisa o atendimento dos requisitos levantados inicialmente e o cumprimento da proposta inicial da ferramenta.

## 6 CONSIDERAÇÕES FINAIS

Neste trabalho são apresentados conceitos de TV digital, um exercício de previsão da demanda futura deste mercado, tecnologias para desenvolvimento Web, e por fim foi proposta e desenvolvida uma ferramenta para autoria de aplicações interativas NCL através de *templates* que faz uso do navegador.

A implementação foi detalhada de maneira a explicar como foi desenvolvido o projeto e, por fim, exemplificada em forma de estudo de caso, para demonstrar os objetivos finais alcançados.

### 6.1 Trabalhos futuros

O aplicativo final, embora bastante funcional, tem espaço para implementação de melhorias. Uma melhoria imediata é a refatoração do código fonte da aplicação.

Outras melhorias viriam da implementação dos requisitos que não puderam ser atendidos completamente:

- A tradução só atingiu a interface. As mensagens das ações (salvar, exportar, etc) só foram implementadas em um único idioma (inglês) e diretamente no código-fonte;
- Os grids de propriedades de mídias e regiões só permitem a exibição, quando deveriam permitir alteração direta;
- A aplicação foi testada, principalmente, nos navegadores Mozilla Firefox e o Google Chrome. Não foram efetuados testes exaustivos no Internet Explorer, por exemplo.

Além disso, algumas sugestões a serem implementadas futuramente, que acrescentariam mais funcionalidade à ferramenta, são:

- Permitir o *upload* de arquivos de áudio, fornecendo funcionalidade semelhante ao tipo de arquivo de vídeo, isto é, utilizar HTML5 ou Flash de acordo com a necessidade;
- Permitir o *upload* de múltiplos arquivos;
- Fornecer *feedback* visual de porcentagem de *upload* de arquivos;
- Reescrita do código de leitura de *templates* para permitir regiões aninhadas;
- Uso de *templates* TAL (Soares Neto et al., 2010);
- Validar os arquivos de *template* e TIF. Atualmente, assume-se que todos são válidos;
- Permitir a criação de novos *templates* baseados em *templates* existentes;
- Avaliação da ferramenta com os usuários finais.

## 6.2 Conclusões

No desenvolvimento do trabalho proposto, a especificação multimídia do HTML5, apesar de bastante promissora, não se mostrou viável devido ao impasse relacionado ao mpeg4, *codec* essencial no contexto de TV Digital. Assim que esse impasse for resolvido, será interessante atualizar a ferramenta creaTiVe.

A Tabela 6.1 exibe os requisitos que foram atendidos, parcialmente atendidos ou não atendidos.

Tabela 6.1: Situação final dos requisitos levantados

Requisito	Descrição	Situação
RF01	Criação de projeto	Atendido
RF02	Edição de projeto	Atendido
RF03	Gravar projeto	Atendido
RF04	Customizar <i>template</i>	Atendido
RF05	Manipulação de Regiões	Atendido
RF04	Customizar <i>template</i>	Atendido
RF05	Manipulação de Regiões	Atendido
RF06	Arrastar e soltar	Atendido
RF07	Mudança de idioma	Atendido parcialmente
RF08	Upload de arquivos	Atendido
RF09	Grid de propriedades	Atendido parcialmente
RF10	Exportar projeto	Atendido
RNF01	Código-fonte aberto	Atendido
RNF02	Repositório	Atendido
RNF03	Compatibilidade entre navegadores	Atendido parcialmente

Com base na tabela 6.1, conclui-se que a ferramenta atendeu a praticamente tudo o que fora proposto e pode vir a ser um excelente recurso para criadores de aplicações interativas Ginga-NCL, principalmente por conta da facilidade de uso e interface intuitiva.

O código-fonte da ferramenta creaTiVe encontra-se disponível no endereço:

<http://code.google.com/p/creative-ncl/>.

## Referências Bibliográficas

- Azevedo RGA, Teixeira MM, Soares Neto CS. NCL Eclipse: Ambiente Integrado para Desenvolvimento de Aplicações para TV Digital Interativa em Nested Context Language. SBPC 2009; IV: 1 a 19.
- Castledine E, Sharkie C. JQUERY Novice to Ninja. New Kicks and Tricks. Second edition. Sitepoint, 2012.
- Clube-NCL. Disponível em <<http://clube.ncl.org.br>>, 2012. Acesso em 07/01/2012.
- Ginga-NCL. Disponível em <<http://www.gingancl.org.br>>, 2012. Acesso em 12/01/2012.
- Guimarães RL. Composer: um ambiente de autoria de documentos NCL para TV digital interativa. Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro, 2007.
- HTML. Histórico das linguagens HTML e XHTML. Disponível em <<http://www.referenciando.com/referencias/html-xhtml/html-historia>>, 2012. Acesso em 11/03/2012.
- IBGE. Indicadores do período de 2004 a 2009. Disponível em <<http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2009/comentarios200.9.pdf>>, 2009. Acesso em 23/02/2012.
- IBOPE. Investimentos em mídia crescem 16% em 2011. Disponível em <<http://www.ibope.com.br/calandraWeb/servlet/CalandraRedirect?temp=5&proj=PortallBOPE&pub=T&db=caldb&comp=Not%EDcias&docid=0BC9A850D6051F458325798A006A0103>>, 2012. Acesso em 25/02/2012.
- JSRender. Disponível em <<https://github.com/BorisMoore/jsrender/>>, 2012. Acesso em 09/03/2012.
- Kessin Z. Programming HTML5 Applications. 1st edition. O'Reilly, 2012.

Loudon K. Developing Large Web Applications. 1st edition. O'Reilly, 2010.

Modernizr. Disponível em <<http://modernizr.com>>, 2012. Acesso em 15/05/2012.

Oliveira CCN. TVision - Ferramenta Gráfica para desenvolvimento de aplicações para TV Digital no formato Ginga-NCL. Trabalho de graduação, Centro de Informática - Universidade Federal de Pernambuco, 2009.

Pfeiffer S. The Definitive Guide to HTML5 Video. 1st edition. O'Reilly, 2010.

PHP. Histórico. Disponível em <[http://www.php.net/manual/pt\\_BR/history.php.php](http://www.php.net/manual/pt_BR/history.php.php)>, 2012. Acesso em 11/02/2012.

Sarrion E. JQuery UI. 1st edition. O'Reilly, 2012.

Soares LFG, Barbosa SDJ. Programando em NCL 3.0. 2nd edition. PUC-Rio, 2012.

Soares Neto CS, Soares LFG, Souza CS. TAL - Linguagem para Autoria de Templates de Documentos Hipermedia. WebMedia 2010; .

Terra. Governo obriga uso do Ginga: TV interativa depende da indústria. Disponível em <<http://tecnologia.terra.com.br/noticias/0,,OI5669999-EI12882,00-Governo+obriga+uso+do+Ginga+TV+interativa+depende+da+industria.html>>, 2012. Acesso em 12/01/2012.

VideoJS. Disponível em <<http://videojs.com/>>, 2012. Acesso em 22/04/2012.