

Universidade Federal de Santa Catarina
Ciências da Computação

Juliano de Souza Krieger

***Sistema de Apoio a Criação de Interatividade para
Televisão Digital***

Orientador:
Prof. Dr. rer.nat. Aldo von Wangenheim

Co-orientador:
Prof. Mathias Henrique Weber

Florianópolis, Maio de
2010

Resumo

Este trabalho apresenta uma pesquisa na área de televisão digital com ênfase em ferramentas de autoria. O padrão brasileiro de televisão digital é estudado e documentado. É realizado um levantamento das ferramentas disponíveis para autoria em televisão digital no mercado e na bibliografia. A partir da pesquisa e com o apoio de uma equipe de pesquisa na área, melhorias em questões de usabilidade e novas funcionalidades são propostas e implementadas para o protótipo inicial de uma ferramenta de autoria.

Palavras-chaves: TV Digital, Ferramentas de Autoria, Interatividade.

Abstract

This work presents a research in the field of digital television with an emphasis on authoring tools. The Brazilian digital television standard is studied and described. A survey on authoring tools available in the market and in the literature is conducted. Based on this research and on the support of a research team, improvements in usability-related matters and also new features are proposed and implemented in a prototype authoring tool.

Keywords: Digital TV, Authoring Tools, Interactivity.

Sumário

Sumário.....	4
1 Introdução.....	9
1.1 Contextualização.....	9
1.2 Objetivos.....	9
1.2.1 Geral.....	9
1.2.2 Específicos.....	9
1.3 Justificativa.....	10
1.4 Metodologia.....	10
1.5 Organização.....	10
2 Usabilidade: Aspectos Conceituais e Gerais.....	12
3 Padrão Brasileiro de Televisão Digital.....	13
3.1 O Middleware Ginga-J.....	13
3.2 O Middleware Ginga-NCL.....	13
3.2.1 NCLua.....	14
3.2.2 O Ambiente de Testes.....	14
4 Ferramentas de Autoria.....	15
4.1 Icareus iTV Suite Author.....	15
4.2 JAME Author.....	16
4.3 AltComposer.....	16
4.4 GRiNS.....	17
4.5 LimSee2.....	17
4.6 Composer.....	18
5 Melhorias Propostas para uma Ferramenta de Autoria com Foco em Conteúdo Jornalístico.....	20
5.1 A Ferramenta de Autoria Desenvolvida - Célula.....	20
5.2 Protótipo Inicial.....	20
5.3 Melhorias Propostas.....	22
6 Melhorias Implementadas no Célula.....	24
6.1 Pré-visualização.....	25
6.2 Editor de regiões.....	26
6.3 Definição de plugins para modelos de interatividade.....	27

6.4 Modelagem dos aplicativos interativos.....	31
6.5 Geração de Programas com aplicativos Interativos no padrão Ginga-NCL.....	35
6.6 Internacionalização do Célula.....	37
7 Modelos de Interatividade Criados para a Ferramenta.....	39
7.1 Jogo de Memória.....	39
7.2 Informação Textual.....	40
7.3 Jogo de Perguntas.....	41
7.4 Leitor de Notícias em RSS.....	42
7.5 Galeria de Imagens.....	43
8 Conclusões.....	45
8.1 Trabalhos futuros.....	45
Referências Bibliográficas.....	47

Índice de figuras

Figura 1: Icareus iTV Suite Author.....	16
Figura 2: LimSee2.....	17
Figura 3: Composer.....	18
Figura 4: As quatro visões do Composer.....	19
Figura 5: Protótipo inicial do Célula.....	21
Figura 6: Célula na versão atual.....	24
Figura 7: Pré-visualização na primeira versão do Célula.....	25
Figura 8: Pré-visualização no Célula.....	26
Figura 9: Editor de regiões do primeiro protótipo do Célula.....	26
Figura 10: Editor de Regiões no Célula.....	27
Figura 11: Elemento "aplicativoTVDI".....	28
Figura 12: Elementos básicos.....	28
Figura 13: Atributos específicos do Jogo de Perguntas.....	29
Figura 14: Arquivos para o Leitor de Notícias.....	30
Figura 15: Repositório de Interatividades.....	31
Figura 16: Diagrama de Classes em UML de parte do Célula.....	31
Figura 17: Propriedades do modelo de interatividade "Jogo de Perguntas" no primeiro protótipo do Célula.....	33
Figura 18: Atributos do modelo "Jogo de Perguntas" no protótipo inicial do Célula.....	34
Figura 19: Painel de propriedades do "Jogo de Perguntas".....	34
Figura 20: Representação em XML dos atributos do painel de propriedades da figura 19.....	34
Figura 21: Elemento "object" do arquivo de parâmetros.....	36
Figura 22: Grupo de parâmetros.....	36
Figura 23: Linha adicionada no arquivo Lua pelo Célula.....	36
Figura 24: Internacionalização no Célula.....	37
Figura 25: Aplicativo gerado pela ferramenta a partir do modelo “Jogo de Memória” .	40
Figura 26: Aplicativo gerado pela ferramenta a partir do modelo “Informação Textual”	41
Figura 27: Aplicativo gerado pela ferramenta a partir do modelo “Jogo de Perguntas”.	42
Figura 28: Aplicativo gerado pela ferramenta a partir do modelo “Leitor de Notícias”.	43

Figura 29: Telas de uma galeria de imagens criada a partir do modelo "Galeria de Imagens".....	44
--	----

Lista de abreviaturas e siglas

HDTV	High-Definition Television
ISDB-T	Integrated Services Digital Broadcasting-Terrestrial
SBTVD	Sistema Brasileira de Televisão Digital
SDTV	Standard-Definition Television
SINRED	Sistema Nacional de Radiofusão Educativa
CPqD	Centro de Pesquisa e Desenvolvimento em Telecomunicações
TVDi	Televisão Digital Interativa
UHF	Ultra High Frequency
URD	Unidade Receptora-Decodificadora
VHF	Very High Frequency
DVB	Digital Video Broadcasting
XML	Extensible Markup Language
MHP	Multimedia Home Platform
GEM	Global Executable MHP

1 Introdução

1.1 Contextualização

Com o início da transmissão digital no Brasil, muitas melhorias são apresentadas para os usuários do meio de transmissão televisivo, sendo as principais características a alta resolução e a possibilidade de interatividade com o telespectador.

A televisão digital no Brasil surge com o desenvolvimento de uma nova camada de software, chamada de *middleware*, feita em território nacional e já com muita aceitação mundialmente. Com o novo *middleware*, surge a questão da criação de conteúdo interativo para o padrão recém criado.

No âmbito da criação de programas para televisão digital, é inviável e impraticável que seja necessário um programador para qualquer aplicação que se queira desenvolver, neste contexto, tornam-se necessárias as ferramentas de autoria para televisão digital, que são softwares com o objetivo de gerar conteúdo para uma futura transmissão.

1.2 Objetivos

1.2.1 Geral

A partir de um protótipo de uma nova ferramenta de autoria, propor e implementar melhorias e novos modelos de interatividade para a mesma.

1.2.2 Específicos

- Documentar o atual estado do padrão brasileiro de transmissão;
- Documentar uma forma de gerar programas interativos para o padrão brasileiro, padronizada que permita a automação de geração;
- Propor e implementar melhorias propostas para a ferramenta;
- Propor e implementar modelos de interatividade para a ferramenta;

- Criar relatórios e tutoriais com o objetivo de popularizar a ferramenta de autoria..

1.3 Justificativa

O Sistema Brasileiro de Televisão Digital (SBTVD) foi criado com o objetivo de ser amplamente difundido. Para a ampla divulgação da televisão digital no Brasil, cada vez mais se observa a inviabilidade da constante necessidade de um programador para a criação de programas. Para ampliar o público da televisão é necessário aumentar a velocidade de criação de programas, para isto, a automação deste processo se faz necessária. A ferramenta de autoria tem este objetivo.

Com esta justificativa a proposta de melhorias para um software que tem por objetivo facilitar a criação de programas para televisão digital se mostrou um excelente tema de pesquisa. Utilizar este software para ampliar a divulgação e utilização do Sistema Brasileiro de Televisão Digital é um dos fatores que mais entusiasma a pesquisar as melhores maneiras de se produzir conteúdo.

1.4 Metodologia

A metodologia será analítica e prática. A partir de um protótipo do sistema, será feita uma bateria de testes a fim de o próprio usuário final propor maneiras de deixar o sistema mais usual e intuitivo. Estudar também outros sistemas que tem a mesma finalidade tanto no mercado quanto na literatura científica para identificar funcionalidades que são importantes para propor ao sistema.

1.5 Organização

Este trabalho foi organizado da seguinte maneira: No próximo capítulo é feita uma revisão bibliográfica sobre usabilidade com foco em qualidade de software. O capítulo 3 faz uma introdução histórica sobre as mídias televisivas no território brasileiro. No capítulo 4 o padrão brasileiro de televisão digital e seus *middlewares* são introduzidos. No capítulo 5 é apresentada uma revisão das ferramentas de autoria disponíveis no mercado e na bibliografia. O capítulo 6 apresenta propostas de melhorias para a ferramenta com base na pesquisa realizada. No capítulo 7, a partir das propostas do

capítulo 6, as melhorias implementadas são apresentadas e a versão atual da ferramenta é exposta. Por fim o capítulo 9 apresenta as conclusões do trabalho feito e trabalhos futuros são propostos.

2 Usabilidade: Aspectos Conceituais e Gerais

O termo usabilidade é recente e muito discutido em diversas áreas de conhecimento especialmente na ciência da computação e na relação homem-máquina.

Conforme a norma ISO 9241-11 (1998), usabilidade é a “capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”.

A Usabilidade e a questão de qualidade de software são assuntos que mantêm estreita relação, de acordo com a autora Dias (2003) A qualidade de um software depende de fatores como funcionalidade, confiabilidade, usabilidade, eficiência, possibilidade de manutenção, e portabilidade. A usabilidade tem muita influência na qualidade quando o usuário do software é levado em conta, alguns princípios são: satisfação subjetiva, facilidade de aprendizado, facilidade de memorização e baixa taxa de erros.

É de extrema importância que um software ou produto que será utilizado por terceiros tem de ter alta usabilidade. Quando é dada a devida importância para a interface gráfica e a arquitetura da informação, a usabilidade aumenta e a relação do usuário com o software melhora significativamente, de forma confortável, acessível e intuitiva, aumentando também a produtividade do sistema

3 Padrão Brasileiro de Televisão Digital

A camada de software que permite a criação de aplicações interativas para o padrão brasileiro de televisão digital, chamado de Ginga, possui dois subsistemas: o Ginga-NCL, que utiliza a linguagem NCL (Nested Context Language); e o Ginga-J que utiliza a tecnologia Java para o desenvolvimento das interatividades, tecnologia esta que já é utilizada em outros padrões para TVD ao redor do mundo. Nas próximas sessões os *middlewares* são melhor detalhados.

3.1 O Middleware Ginga-J

O Brasil definiu o Ginga-J para ser o ambiente procedural do padrão brasileiro. A linguagem Java é utilizada para a criação das aplicações e o padrão GEM é utilizado como base para a especificação das APIs a serem utilizadas. Este padrão foi finalmente especificado apenas em abril de 2010, por isto e pela falta de ferramentas para desenvolvimento, neste trabalho ele não foi muito aprofundado.

3.2 O Middleware Ginga-NCL

No middleware Ginga-NCL, é utilizada a linguagem NCL. NCL é uma linguagem declarativa baseada no modelo conceitual NCM (Nested Context Model) que serve para sincronizar mídias no tempo e no espaço em uma aplicação para TVD. Um programa para televisão digital tem como elemento principal um documento NCL, o qual define uma mídia principal que será exibida, geralmente o vídeo principal em tela cheia, e também outras mídias.

A linguagem NCL é muito poderosa, porém quando se pretende fazer algum processamento mais complexo, como fazer animações, algum processamento com texto ou ainda alguma interatividade mais elaborada, utilizando o controle remoto e o canal de retorno por exemplo, ela deixa a desejar. Por ser uma linguagem de declarativa, NCL não foi feita para tratar a parte lógica do programa, para solucionar este problema, foi definido no SBTVD que é possível adicionar objetos de mídia na linguagem Lua.

3.2.1 NCLua

Lua é uma linguagem de programação imperativa, interpretada, leve e dinamicamente tipada, além de ter sido desenvolvida em território nacional. Por estes e outros motivos, foi escolhida para fazer parte do padrão brasileiro de televisão digital. Um objeto de mídia que utiliza esta linguagem é chamado de NCLua.

No Ginga-NCL, é definido um conjunto de quatro bibliotecas adicionais disponíveis nos scripts Lua:

- event – permite a comunicação entre o script Lua e o ambiente;
- canvas – oferece uma API com funções gráficas de desenho e manipulação de imagens;
- settings – exporta uma tabela com variáveis de ambiente e outras definidas pelo autor do documento NCL;
- persistent – exporta uma tabela com variáveis persistentes, disponíveis apenas para objetos imperativos.

3.2.2 O Ambiente de Testes

Para testar as aplicações desenvolvidas foi utilizado o Ginga-NCL Virtual STB [Ginga-NCL, 2009], que é a uma máquina virtual linux para VMWare [VMWare, 2009], configurada com uma implementação de referencia do middleware Ginga-NCL, desenvolvida pela PUC do Rio de Janeiro. VMWare é um software de virtualização de máquinas, desenvolvido por uma empresa de mesmo nome.

Para executar uma aplicação é necessário acessar a máquina virtual via SSH (Secure Shell), transferir todos os arquivos do programa e rodar um *script* que executa o programa a partir do documento NCL principal.

4 Ferramentas de Autoria

GUIMARÃES (2007) explica que uma ferramenta de autoria é empregada com a finalidade de abstrair alguns aspectos, geralmente técnicos, na fase de concepção de uma tarefa. Em outras palavras, permite ao usuário, representar suas ideias em um nível mais alto de abstração, diminuindo assim, a complexidade.

Ferramentas de autoria são diversas e existem para várias finalidades. No âmbito da televisão digital, para WEBER (2009) uma ferramenta de autoria para TVDi deve permitir ao desenvolvedor reunir diferentes elementos de mídia em uma linha de tempo, com a opção de adicionar interatividade entre os elementos .

Este capítulo faz um levantamento das ferramentas existentes hoje explicitando suas principais características. As próximas seções apresentarão as ferramentas de autoria para televisão digital que foram analisadas.

4.1 Icareus iTV Suite Author

A Icareus iTV Suite é uma ferramenta de autoria desenvolvida pela empresa Icareus Technologies. O software é desenvolvido em java com suporte aos seguintes padrões ou plataformas: MHP, OCAP / tru2way, IDway-J and Blu-ray. A partir do site da Cardinal Systems, clicando no link que aponta para a ferramenta Cardinal Studio Professional, o site é redirecionado para o site do Icareus. Portanto supõe-se que o Icareus iTV Suite Author seja o substituto direto para o Cardinal Studio Professional.

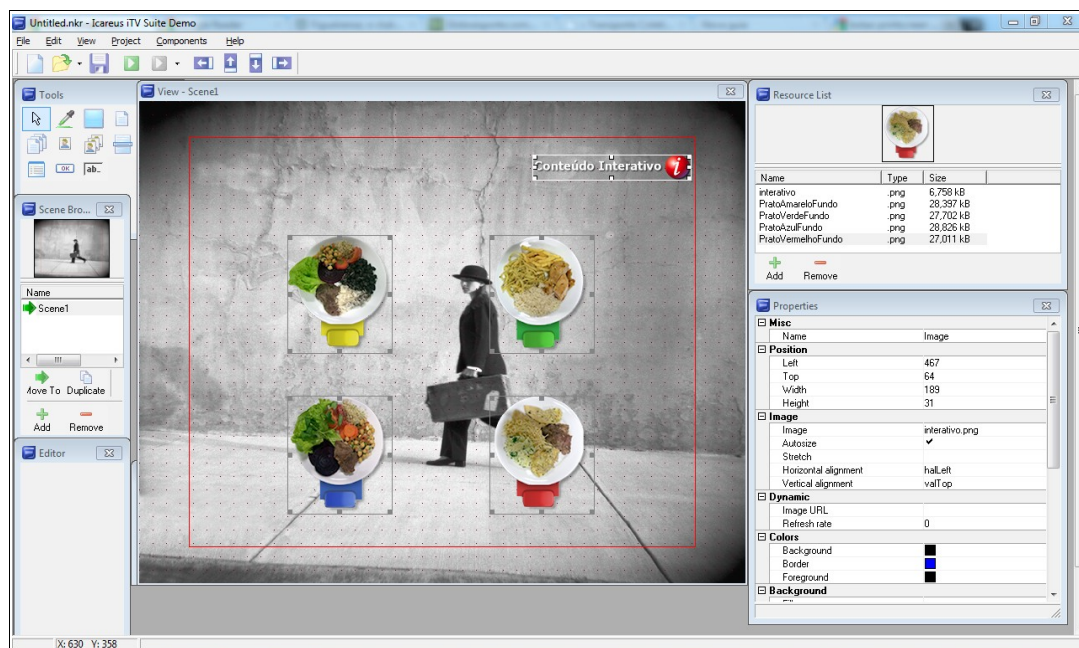


Figura 1: Icareus iTV Suite Author

A partir da página do Icareus iTV Suite Author é possível se registrar e realizar o download de uma versão de demonstração do software. A versão de demonstração disponível na página do desenvolvedor tem o *layout* mostrado na Figura 1.

4.2 JAME Author

O Ambiente JAME Author [Fraunhofer, 2009] é outro projeto descontinuado que criava programas para televisão digital no padrão MHP/OCAP (Open Cable Specification). Continha um emulador MHP para teste e era integrado com uma infraestrutura de broadcast.

De acordo com o site do desenvolvedor, a partir do dia primeiro de outubro de 2009 o software foi descontinuado.

4.3 AltComposer

A AltComposer é uma ferramenta de autoria para criar aplicativos compatíveis com o padrão DVB-MHP. A partir de uma interface gráfica, a ferramenta permitia tanto

usuários leigos criarem aplicações, quanto usuários experientes em programação criarem componentes personalizados (Guimarães, 2007).

É um software que atualmente está descontinuado e existem poucas informações sobre ele na página do fabricante. O AltiComposer era fabricado pela empresa AltiCast (Alticast, 2009), especializada em soluções para televisão.

4.4 GRiNS

SMIL (Synchronized Multimedia Integration Language) é uma linguagem de marcação em XML para sincronização e apresentação de mídias na internet. GRiNS é uma ferramenta de autoria para SMIL. O software possui três visões para auxiliar o usuário na criação: temporal, espacial e textual (GUIMARÃES, 2007).

4.5 LimSee2

A LimSee2 é uma ferramenta de autoria para SMIL, assim como o GRiNS. É um software multi-plataforma, desenvolvido na linguagem Java (SUN, 2009), que pela página do desenvolvedor já está pronto. A visão geral é mostrada na Figura 2.

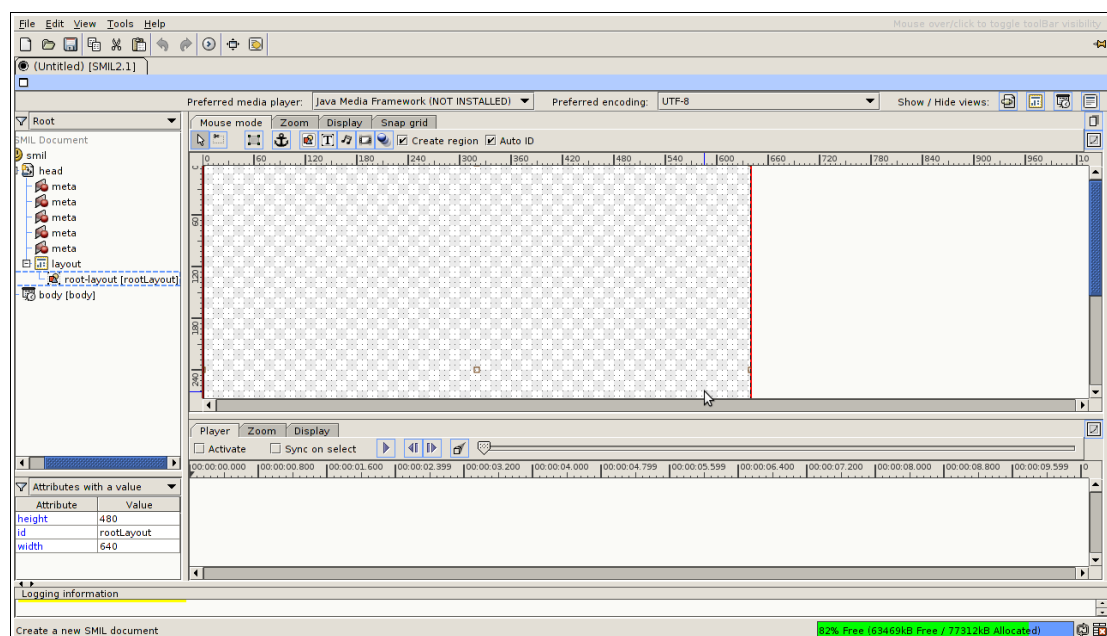


Figura 2: LimSee2

4.6 Composer

NCL (*Nested Context Language*) é uma linguagem declarativa para sincronização de mídias. É a linguagem declarativa definida para o Ginga-NCL, um dos middlewares do Sistema Brasileiro de Televisão Digital. Composer (GUIMARÃES, 2007) é atualmente a única ferramenta de autoria feita no Brasil para edição e criação de documentos NCL.

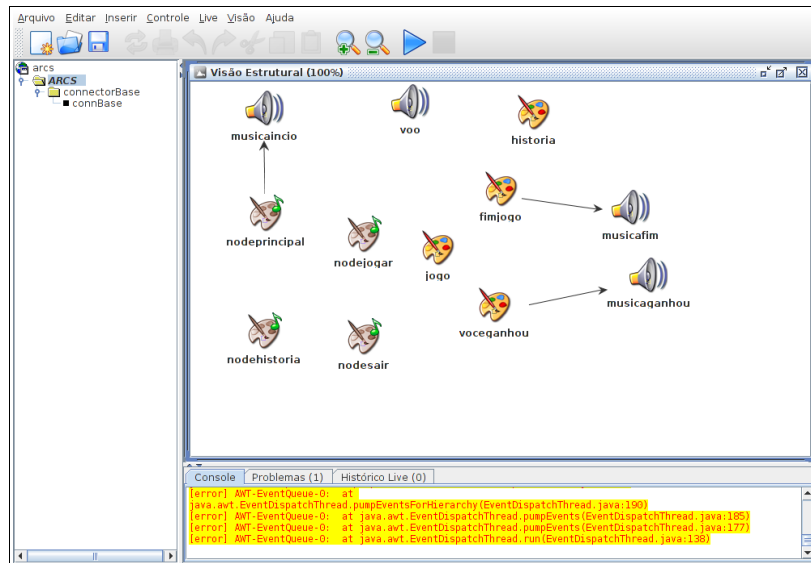


Figura 3: Composer

O Composer (Figura 3) é dividido em quatro visões. Visões são formas de visualizar e editar, ou seja, diferentes abstrações para editar o mesmo programa. As visões são: textual, estrutural, *layout* e temporal. Cada uma delas oferece facilidades para utilizar certas funcionalidades em relação a outras, podendo ser necessário mais de uma visão para conseguir o resultado desejado. Além das visões, o Composer também oferece um simulador, para que o usuário possa testar a aplicação.

A visão textual serve para editar o arquivo NCL manualmente através do texto. É usada quando as outras visões não conseguem expressar tudo que o autor precisa. A visão estrutural facilita ao autor ter uma estrutura lógica do programa. Mídias e eventos são expressos através de um grafo com nós (mídias) e elos (eventos). Os objetos de mídia são apresentados em uma região específica, a visão de *layout* facilita a inserção de regiões para mídias já definidas. A visão temporal expressa os objetos de mídia através de uma linha de tempo, com as condições de transição, para que as mídias sejam apresentadas. A Figura 4 mostra um exemplo de cada uma das quatro visões.

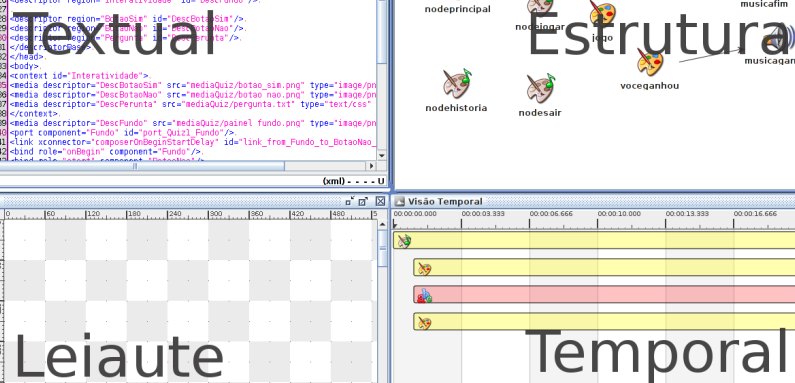


Figura 4: As quatro visões do Composer

5 Melhorias Propostas para uma Ferramenta de Autoria com Foco em Conteúdo Jornalístico

5.1 A Ferramenta de Autoria Desenvolvida - Célula

O Célula consiste em um ambiente baseado em uma linha temporal. Esta linha é determinada a partir da duração de um vídeo, definido como principal. É permitido ao usuário do software adicionar mídias ao programa, baseado no tempo do vídeo principal. Mídias, de acordo com o *middleware* Ginga, podem ser imagens, áudios e vídeos, além de aplicativos interativos, que no caso do Ginga-NCL pode ser na linguagem Lua. O software, atualmente chamado de *Célula* utiliza o conceito de *plugins*, onde apenas um arquivo em XML define um aplicativo interativo.

5.2 Protótipo Inicial

No protótipo inicial da ferramenta de autoria (Figura 5) a interface tinha um menu superior, com botões para criar, abrir e salvar projetos, gerar a aplicação criada e um botão para conseguir informações sobre a aplicação. Abaixo do menu tinha uma biblioteca de mídias, dividido em vídeos, imagens, áudios, e aplicativos. Ao lado da biblioteca havia um painel onde as propriedades da mídia selecionada podiam ser modificadas. Do lado direito do painel de propriedades havia um painel para modificar a região da mídia selecionada. No canto direito era possível observar um painel com uma pré-visualização da mídia selecionada. No painel inferior do software uma linha do tempo mostrava quando cada mídia estaria disponível para o telespectador.



Figura 5: Protótipo inicial do Célula

O Célula nesta versão funcionava da seguinte maneira: o usuário clicava no botão que adicionava mídias na biblioteca, e cada mídia incluída aparecia automaticamente na linha de tempo, sendo que o primeiro vídeo adicionado era automaticamente definido como vídeo principal do programa. As cores de cada mídia era determinada pelo seu tipo, vídeo em azul, áudio em vermelho, imagem em verde e aplicativos na cor rosa, e a cor na linha de tempo era definida da mesma forma. Quando uma mídia era selecionada na biblioteca, as propriedades da mídia apareciam no respectivo painel.

Esta versão do software tinha muitos problemas de estabilidade e usabilidade, além de alguns problemas de projeto. O painel de propriedades mostrava todos os atributos em forma de texto, o que poderia gerar erros ao modificar. O painel para modificar a região da mídia não era muito intuitivo e muitas vezes cálculos errados eram feitos, dependendo do movimento do mouse. Além dos detalhes de interface e usabilidade, outras questões ainda precisavam ser definidas, principalmente na criação e geração dos aplicativos com interatividade.

5.3 Melhorias Propostas

Foi feito um levantamento das ferramentas de autoria para televisão digital existentes no mercado e uma pesquisa das necessidades de jornalistas observadas

durante entrevistas. As entrevistas foram feitas com um profissional do ramo com experiência em produção de conteúdo televisivo. O professor e jornalista Fernando Crocomo, com doutorado na área de televisão digital foi um dos profissionais que mais colaboraram com o projeto. A partir disto e com o apoio de uma equipe de pesquisa em televisão digital, uma equipe de design, experiência no desenvolvimento de software e do protótipo inicial da ferramenta, foram propostas melhorias para o software, abordando, tanto questões de usabilidade, quanto questões internas do próprio projeto do software.

Para questões de interface e design o projeto recebeu propostas de uma empresa especializada em design. O logotipo do software, algumas questões de cores e o *layout* do software foram propostos por esta empresa parceira. Muitas ideias sobre funcionalidades que o software precisaria ter foram feitas pelo profissional de jornalismo.

Propostas de melhorias:

- Uma funcionalidade que foi observada também em outras ferramentas de autoria foi a pré-visualização do programa de acordo com a linha de tempo e as mídias habilitadas para o instante selecionado. Na primeira versão do software já havia uma pré-visualização, mas apenas mostrava a mídia selecionada no painel da biblioteca. A proposta é mostrar na janela de pré-visualização além do momento do vídeo principal no instante selecionado, também as mídias que estariam habilitadas naquele instante.
- O protótipo inicial da ferramenta já continha um editor de regiões para as mídias. Este editor, entretanto, tinha suas funcionalidades muito limitadas, o que impedia o usuário a fazer uma modificação da região com mais precisão ou visualizar a imagem da mídia selecionada, além de ser difícil de usar. Foi proposto um editor de regiões com melhores funcionalidades e com melhor usabilidade.
- Uma das propostas da ferramenta é gerar programas com aplicativos interativos. No primeiro protótipo já havia o conceito de documentos XML para definir um aplicativo, mas o usuário precisava clicar em adicionar e procurar o arquivo correspondente à interatividade escolhida. A proposta neste caso foi continuar com os documentos XML para aplicativos interativos mas incluir o conceito de

plugins, onde as interatividades seriam incluídas automaticamente e encontradas facilmente pelo usuário.

- O documento XML que define um aplicativo interativo era restritivo e impedia que o desenvolvedor criasse modelos de interatividade mais complexos. Para este problema, a modelagem de um aplicativo interativo teve de ser repensada. Foi importante criá-la de forma abrangente, de forma que pudesse ser estendida de acordo com a evolução do software, além de conseguir satisfazer todas as exigências dos usuários. Neste ponto foi necessário modificar além da modelagem interna no Célula, também a maneira como os modelos são definidos através do documento XML e a forma como as interatividades são geradas a partir desta nova modelagem.
- Pelo software ter um dos objetivos de gerar aplicações para múltiplos padrões (pelo menos os dois *middlewares* brasileiros), para cada padrão há um gerador. Portanto, definida a modelagem e o funcionamento das interatividades nas aplicações, houve a necessidade de adaptar os geradores das aplicações. O foco maior foi com os modelos de interatividade no padrão Ginga-NCL pois neste momento é o único bem definido e com ferramentas para teste e validação no Brasil.
- Qualquer software que tem o objetivo de ser utilizado por terceiros deveria ter suporte a múltiplos idiomas. No Célula foi importante a implementação da internacionalização.

6 Melhorias Implementadas no Célula

O Célula, na versão atual conta com muitas melhorias implementadas, atualmente o software conta com o *layout* mostrado na Figura 6. Visualmente o sistema conta com um menu na parte superior em mais quatro painéis diferenciados: "Biblioteca", "Visualização", "Propriedades" e "Linha do tempo" na região inferior. Nas próximas sessões serão dados maiores detalhes sobre cada melhoria implementada.

Uma das principais diferenças com relação à primeira versão é a forma como o Célula trabalha com as mídias adicionadas ao programa. Diferente da versão anterior, agora ao adicionar uma mídia na biblioteca, ela não entra automaticamente na linha de tempo do programa, para isto, é necessário que o usuário clique na mídia incluída na biblioteca, arraste até a linha de tempo e solte na posição escolhida. Para definir um dos vídeos como principal, o usuário apenas precisa arrastá-lo até a primeira posição da linha de tempo.

A linha de tempo, além de mostrar o intervalo de tempo quando a mídia estará disponível, também indica a cor do botão a ser selecionado para lançá-la. A linha de tempo também permite editar o instante e o intervalo que a mídia selecionada estará disponível, apenas modificando o tamanho da barra colorida com o *mouse*.

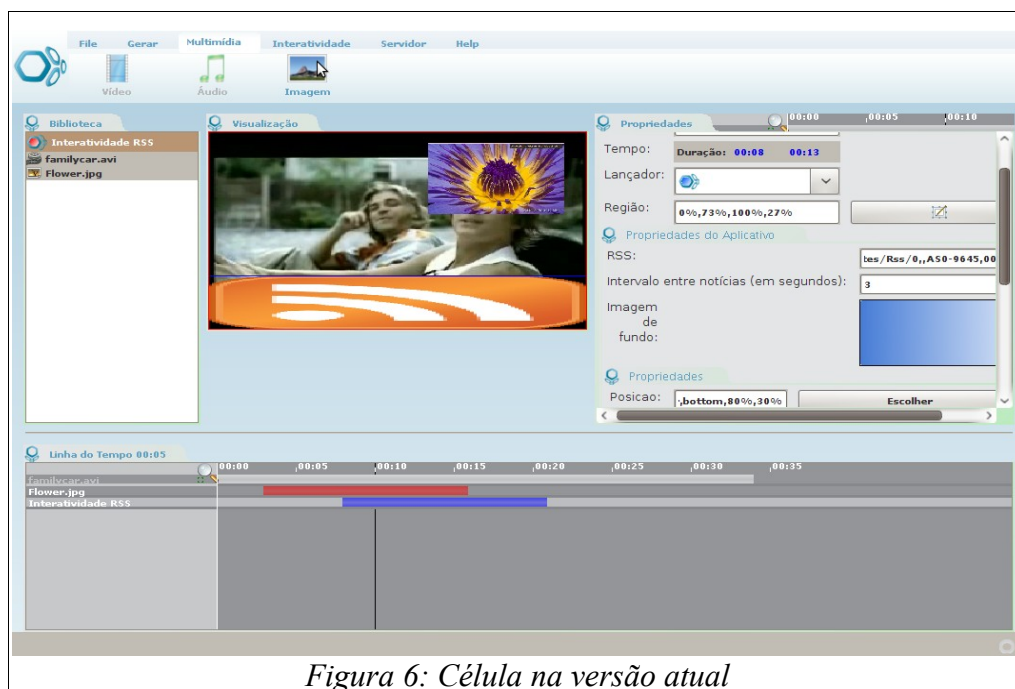


Figura 6: Célula na versão atual

6.1 Pré-visualização

Esta função já existia no protótipo, mas apenas mostrava a mídia selecionada na biblioteca, como na Figura 7, onde um vídeo estava selecionado. A vantagem desta versão é que era possível tocar o vídeo, funcionalidade não permitida na versão atual. A desvantagem é a impossibilidade de visualizar as mídias de acordo com o tempo e suas regiões no vídeo principal.



Figura 7: Pré-visualização na primeira versão do Célula

Foi considerado que visualizar o vídeo, que pode ser tocado em qualquer outro *player* de vídeo, era menos importante que conseguir visualizar como as mídias estariam dispostas no decorrer do programa. Após a implementação desta melhoria, quando o usuário clica em algum ponto da linha de tempo, é lançado um evento que chama uma função que calcula o instante selecionado no vídeo principal, extraíndo a imagem correspondente a este instante. Então todas as mídias são varridas e são extraídas aquelas que naquele instante de tempo estariam habilitadas. O que é desenhado na janela de pré-visualização é a imagem extraída do vídeo, e em uma camada acima, são desenhadas as mídias habilitadas naquele instante, levando em conta sua região.

Na Figura 8 é possível ver uma imagem do vídeo principal ao fundo, e por cima do vídeo é possível ver mais duas mídias naquele instante, uma imagem e uma interatividade. As mídias já são apresentadas em suas respectivas regiões.

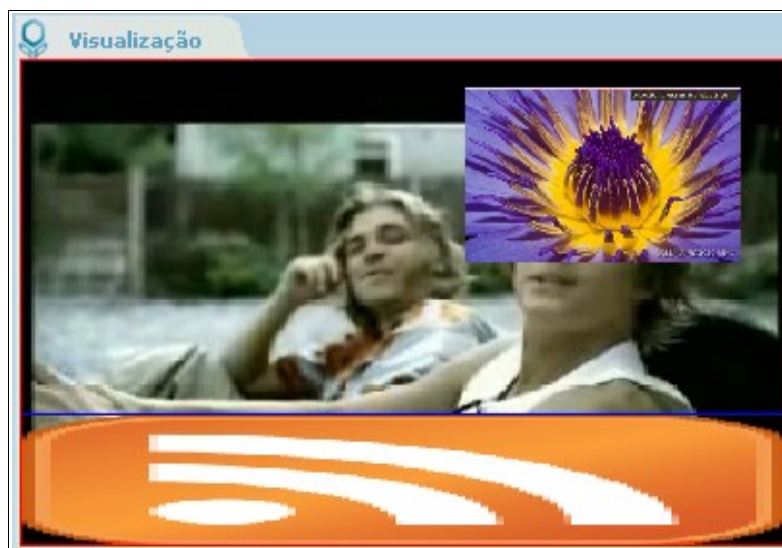


Figura 8: Pré-visualização no Célula

6.2 Editor de regiões

Havia um editor de regiões que estava sempre visível na *interface* do Célula, Figura 9. O canto superior esquerdo representava a origem (0,0) no eixo X e Y e o canto inferior direito representava o limite da região disponível (100%,100%). Quando o usuário clicava em qualquer posição do painel, um novo retângulo era criado para determinar a região da mídia selecionada.

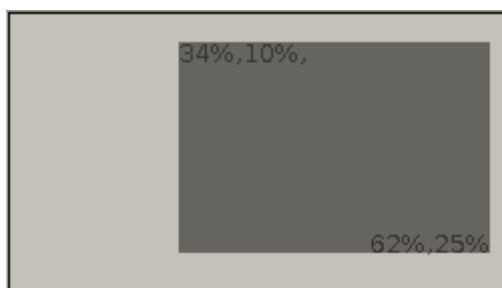


Figura 9: Editor de regiões do primeiro protótipo do Célula.

As melhorias foram implementadas e atualmente no painel onde se edita as propriedades da mídia selecionada há um campo onde mostra a região e ao lado um botão que abre a janela para edição da região da mídia. Botão e janela são mostrados na Figura 10.

O editor de regiões conta com dois painéis, do lado esquerdo mostra a região visualmente, com a imagem da mídia sendo editada e com possibilidade de edição com o *mouse*, modificando a posição da mídia quando se clica dentro do retângulo, ou modificando o tamanho, quando se clica nas arestas ou nos cantos do retângulo. É possível criar uma região inteiramente nova quando se clica fora do retângulo e se arrasta o *mouse* para outra posição. O lado direito da janela conta com campos para se editar numericamente a região, fazendo uma edição mais precisa quando o *mouse* não consegue satisfazer o usuário.

A proposta inicial deste painel era editar a região das mídias adicionadas no programa. Entretanto ele foi implementado de forma que também pode ser usado para editar parâmetros para os aplicativos interativos, como vai ser explicado ao falar sobre as interatividades.

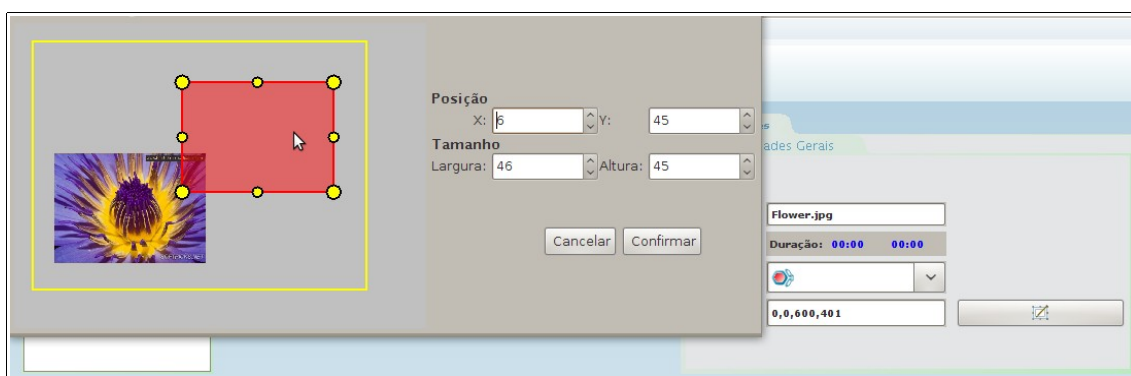


Figura 10: Editor de Regiões no Célula

6.3 Definição de *plugins* para modelos de interatividade

Por ser um dos requisitos iniciais do software, desde o primeiro protótipo da ferramenta as interatividades podiam ser incluídas no programa a partir de um documento XML, porém os usuários tinham que procurar o arquivo com extensão ".xml" para adicionar a interatividade ao programa. O que foi proposto e implementado é o conceito de *plugin*. Os documentos XML que definem cada modelo de interatividade são colocados na pasta de modelos e o software varre todos os arquivos da pasta com extensão ".xml" com a finalidade de criar um repositório de interatividades.

A partir disto foi incluído no menu principal do programa um botão que ao clicar abre uma nova janela com uma lista de interatividades, que não faz mais do que varrer a pasta de modelos e apresentar de forma gráfica e mais agradável ao usuário todas as interatividades disponíveis no repositório. A janela com a lista de interatividades é apresentada na Figura 15.

Pensando nas necessidades dos jornalistas e no que permite o padrão brasileiro de Televisão Digital especificamos o arquivo que especifica um modelo de interatividade. Este arquivo se resume em um documento XML que explicita os parâmetros da interatividade e os arquivos necessários para ele funcionar

O arquivo de descrição do modelo começa com o elemento “aplicativoTVDI”, que possui os atributos “preview” e “tipo”. O atributo “preview” indica o caminho para uma imagem que caracteriza o aplicativo, enquanto o atributo “tipo” identifica para quais *middlewares* o modelo está disponível. Exemplo na Figura 11.

```
<aplicativoTVDI preview="Imagens/rss.png" tipo="Ginga-NCL">
```

Figura 11: Elemento "aplicativoTVDI"

Dentro do elemento “aplicativoTVDI” vem os elementos básicos de qualquer elemento de mídia são “tipo”, “nome”, “descricao”, “inicio”, “duracao” e “regiao”. O elemento “tipo” pode receber os seguintes parâmetros: “jornalístico”, “educacao” e “jogos”, é utilizado para diferenciar os aplicativos. O “nome” aparece na lista de modelos disponíveis. Dentro do elemento “descricao” é incluído um texto mais explicativo sobre o aplicativo. Os elementos “inicio”, “duracao” e “regiao” são auto-explicativos e podem ser modificados pelo usuário. A Figura 12 mostra os elementos básicos para o leitor de notícias.

```
<tipo>jornalístico</tipo>
<nome>Interatividade RSS</nome>
<descricao>Leitor de notícias ou agregadores em RSS</descricao>
<inicio>3</inicio>
<duracao>20</duracao>
<regiao>0,0,100%,100%</regiao>
```

Figura 12: Elementos básicos

Depois dos elementos básicos, vem os atributos específicos para cada modelo, que são definidos pelo criador do modelo. Dentro do elemento “atributos” é possível adicionar elementos “item” e “itemGrupo”. Dentro de um “item” é possível adicionar elementos “propriedade”, que no arquivo “objects.xml” do programa gerado entrarão como atributos dos elementos “object”. Um elemento “itemGrupo” é utilizado para adicionar grupos de elementos “item”. Os atributos para o elemento “itemGrupo” são:

- “nome” que será o identificador para o usuário na ferramenta;
- “type” que será passado ao documento de atributos do programa gerado;
- “min” e “max” serve para restringir o usuário da ferramenta a adicionar um número inválido de itens neste grupo.

Para o elemento “item”, que define um parâmetro para o aplicativo, os atributos são: “nome”, “descricao”, “type”, “tipo” e “default”. Estes atributos são melhor explicados na próxima seção. Na Figura 13, os atributos do jogo de perguntas e respostas.

```
<atributos>
  <item nome="Imagem de Fundo" descricao="Defina a imagem de fundo para o Quiz"
    type="fundo" tipo="imagem" default="fundoAzul.png">
    <propriedade nome="Posicao" type="position" tipo="posicao"
      default="center,bottom,90%,73%"/>
    </item>
    <item nome="Posição do Quiz" type="posicaoQuiz" tipo="posicao"
      default="center,bottom,86%,68%"/>
    <item nome="Cor da letra do quiz" type="corTexto" tipo="cor" default="white"/>
    <item nome="Cor da pergunta" type="corQuestao" tipo="cor" default="green"/>
    <item nome="Cor da opção selecionada" type="corOpcao" tipo="cor" default="red"/>
    <item nome="Cor da resposta" type="corResultado" tipo="cor" default="white"/>
    <item nome="Fonte da letra" type="fonte" tipo="fonte" default="vera.ttf"/>
    <item nome="Tamanho da letra" type="fontsize" tipo="numero" default="20"/>
    <itemGrupo nome="Perguntas" type="perguntas" min="1">
      <item nome="Pergunta" type="pergunta" default="Pergunta?" tipo="textarea"/>
      <itemGrupo nome="Alternativas" type="alternativas" min="1" max="7">
        <item nome="Alt" default="Alternativa" type="alternativa" tipo="texto">
          <propriedade nome="Pontuacao" type="pontuacao" tipo="numero"
            default="0"/>
        </item>
      </itemGrupo>
    </itemGrupo>
    <itemGrupo nome="Regras para o resultado das questoes" type="regras">
      <item nome="Regra" type="regra" tipo="textarea">
        <propriedade nome="De" type="de" default="0" tipo="numero"/>
        <propriedade nome="Até" type="ate" default="0" tipo="numero"/>
      </item>
    </itemGrupo>
  </atributos>
```

Figura 13: Atributos específicos do Jogo de Perguntas

O elemento “itemGrupo” serve para adicionar um número variável de argumentos para os parâmetros. Por exemplo as perguntas e as alternativas das perguntas, onde o usuário pode adicionar quantas ele quiser, dependendo do mínimo e máximo estipulado pelo modelo.

Após os atributos vem o elemento “arquivos” onde são incluídos os arquivos necessários para o aplicativo. O elemento que define um arquivo tem como nome “arquivo” e possui os atributos “tipo” e “nome”. O atributo “tipo” define para que *middleware* aquele arquivo precisa ser adicionado, pode receber “Ginga-NCL” ou “DVB”, ou mais de um *middleware* separado por vírgula, como por exemplo “Ginga-NCL,DVB”. Já o atributo “nome” especifica o caminho para o arquivo, que tem como referência a pasta dos modelos. O primeiro arquivo adicionado para o *middleware* “Ginga-NCL” é definido como o aplicativo a ser chamado pelo documento NCL. Como exemplo, o Leitor de Notícias na Figura 14.

```
<arquivos>
  <arquivo tipo="Ginga-NCL" nome="lua/rss.lua"/>
  <arquivo tipo="Ginga-NCL" nome="libs/libxml.lua"/>
  <arquivo tipo="Ginga-NCL" nome="libs/libcptvdi.lua"/>
  <arquivo tipo="Ginga-NCL" nome="libs/ltn12.lua"/>
  <arquivo tipo="Ginga-NCL" nome="libs/io.o"/>
  <arquivo tipo="Ginga-NCL" nome="libs/io.h"/>
  <arquivo tipo="Ginga-NCL" nome="libs/vera.ttf"/>
  <arquivo tipo="Ginga-NCL" nome="libs/tcp.lua"/>
</arquivos>
```

Figura 14: Arquivos para o Leitor de Notícias

Na versão antiga do Célula, o arquivo XML já existia, porém com algumas diferenças. As principais melhorias estão com relação à definição de grupo de parâmetro com o elemento “itemGrupo” e

Dentro do diretório raiz do Célula fica a pasta “modelos”, que é a pasta onde fica o repositório de interatividades. Na pasta de modelos foram criadas pastas para organizar melhor os arquivos das interatividades. Foram criadas as pastas “Imagens” para as imagens padrões dos modelos, “lua” para os arquivos na linguagem Lua das interatividades, “libs” para eventuais bibliotecas usadas nos aplicativos, “mhp” para os arquivos do padrão MHP e “xml” para arquivos XML necessários para alguns aplicativos.

Preview	Nome	Arquivo
	Interatividade RSS	./modelos/LeitorRss.xml
	Informação Textual Simples	./modelos/newInformacaoText...
	Interatividade Quiz	./modelos/NewQuiz2-V2.xml~
	Informação Textual Simples com Imag...	./modelos/newInformacaoText...
	Interatividade Quiz	./modelos/grupoImagensText...
	Jogo de memória	./modelos/newJogoMemoria.xml
	Interatividade RSS	./modelos/LeitorRss.xml~
	Interatividade Quiz	./modelos/NewQuiz2-V2.xml
	Informação Textual Simples	./modelos/newInformacaoText...
	Interatividade Quiz	./modelos/NewQuiz2.xml
	Interatividade Grupo Imagens Texto	./modelos/grupoImagensText...

Figura 15: Repositório de Interatividades

6.4 Modelagem dos aplicativos interativos

Como os modelos de interatividade ganharam novas características na sua definição, como os conceitos de grupo de atributos, tipo de atributo, entre outros, após defini-los como *plugins* foi necessário definir como eles funcionam e são modificados dentro do software, bem como a forma que as interatividades são geradas pela ferramenta. Pensando nas necessidades dos jornalistas e no que permite o padrão brasileiro de Televisão Digital, modelamos o software da maneira como está especificado na Figura 16, no padrão UML.

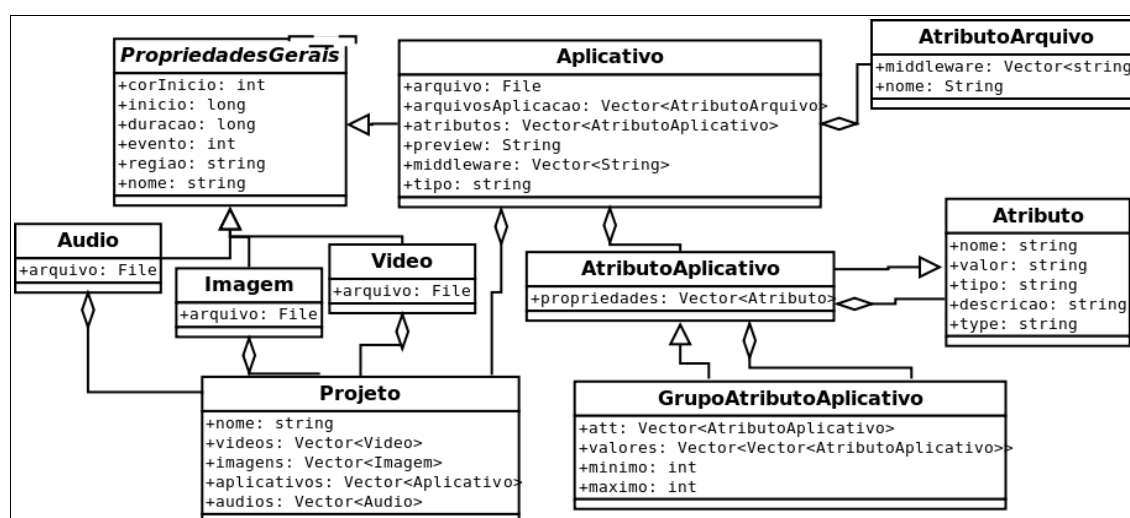


Figura 16: Diagrama de Classes em UML de parte do Célula

Resumidamente um projeto tem uma lista para cada tipo de mídia: áudios; vídeos; imagens e aplicativos. Como o software tem foco em conteúdo jornalístico, o programa sempre tem como base um vídeo principal, que é identificado por ser o primeiro vídeo da linha de tempo e internamente por ser o primeiro vídeo da lista de vídeos. As outras mídias são configuradas no tempo de exibição do tempo do vídeo principal.

Esta modelagem determina que qualquer mídia tem uma cor para início, que corresponde a uma das cores definidas no padrão brasileiro para lançar a mídia. Cada mídia também possui o momento inicial onde a mídia estará disponível e a duração que ela estará disponível. O evento associado, região e nome também são atributos de todas as mídias.

Um aplicativo, além de ter todos os atributos gerais de uma mídia, também possui os atributos específicos do aplicativo, que são definidos pelo modelo. Dos atributos definidos pelo modelo, tem aqueles que podem ser modificados pelo usuário e aqueles que são determinados pelo criador do modelo e não mudam. A imagem de pré-visualização, que aparecerá na janela do repositório de interatividades, o tipo, os *middlewares* disponíveis para aquele aplicativo e os arquivos necessários para rodar a aplicação, que são objetos da classe “AtributoArquivo”.

No documento que define uma interatividade, logo após os atributos gerais de uma mídia, vem os atributos que podem ser modificados pelo usuário, específicos do aplicativo. Na modelagem estes atributos correspondem ao atributo “atributos” da classe “Aplicativo”, que é uma lista de objetos da classe “AtributoAplicativo”. Um objeto desta lista corresponde a um elemento “item” do documento XML, quando o elemento é um “itemGrupo”, é adicionado um objeto da classe “GrupoAtributoAplicativo”. A classe “GrupoAtributoAplicativo” estende a classe “AtributoAplicativo” e possui o atributo “valores”, que é uma lista de listas de objetos da classe “AtributoAplicativo”. “GrupoAtributoAplicativo” foi definida deste jeito para permitir um número variável de entradas para o mesmo atributo.

A classe “AtributoAplicativo” estende a classe “Atributo”, assim todos os objetos da classe “AtributoAplicativo” herdam os atributos da classe “Atributo”, todos do tipo *string*, são eles:

- “nome” - que aparecerá para o usuário da ferramenta como um identificador;

- “descricao” - uma descrição mais extensa sobre o atributo, aparecerá na forma de *tooltip*;
- “valor” - valor do atributo que será passado ao aplicativo na hora de gerá-lo, é o valor que o usuário modifica;
- “tipo” - atributo interno para uso da ferramenta, usado para apresentar o atributo de forma diferenciada, para restringir os erros na modificação do valor. No momento, a ferramenta aceita os tipos “texto”, “textarea”, “numero”, “cor”, “posicao” e “fonte”;
- “type” - que também é de uso interno da ferramenta, este valor será enviado ao aplicativo gerado e é usado para o aplicativo poder identificá-lo.

No documento XML, um elemento “item” pode ter elementos do tipo “propriedade”, por este motivo um objeto da classe “AtributoAplicativo” também possui uma lista de objetos da classe “Atributo”, usado para os atributos que possuem propriedades. Estas propriedades funcionam como um atributo, mas não podem ter outras propriedades dentro.

Na primeira versão do célula, como não havia o conceito de grupo de atributos, o modelo do jogo de perguntas havia uma pergunta, um número fixo de cinco alternativas e um atributo representando a alternativa correta, os atributos tinham apenas dois tipos: “texto” e “textarea”. A figura 17 mostra o painel de propriedades do modelo “Jogo de Perguntas” na versão antiga do Célula enquanto a figura 18 mostra os atributos do modelo no documento XML que o define.

The screenshot shows a configuration panel for a 'Jogo de Perguntas' (Question Game) model. The fields are as follows:

- Nome: Interatividade C
- Inicio(Seg): 5
- Duracao(Seg): 40
- Região: 0.0.100%.100
- Pergunta: Pergunta?
- Alternativa1: Alternativ:
- Alternativa2: Alternativ:
- Alternativa3: Alternativ:
- Alternativa4: Alternativ:
- Alternativa5: Alternativ:
- Alternativacorreta: 4

Figura 17: Propriedades do modelo de interatividade "Jogo de Perguntas" no primeiro protótipo do Célula.

```

<atributos>
  <item nome="pergunta" default="Pergunta?" tipo="textarea"></item>
  <item nome="alternativa1" default="Alternativa 1" tipo="texto"></item>
  <item nome="alternativa2" default="Alternativa 2" tipo="texto"></item>
  <item nome="alternativa3" default="Alternativa 3" tipo="texto"></item>
  <item nome="alternativa4" default="Alternativa 4" tipo="texto"></item>
  <item nome="alternativa5" default="Alternativa 5" tipo="texto"></item>
  <item nome="alternativaCorreta" default="4" tipo="texto"></item>
</atributos>

```

Figura 18: Atributos do modelo "Jogo de Perguntas" no protótipo inicial do Célula.

A Figura 19 mostra como os atributos do aplicativo “Jogo de Perguntas” são mostrados. É possível observar alguns atributos do tipo “cor”, também do tipo “fonte” e “numero”. Também é possível observar como funciona quando há um atributo que representa um grupo de outros atributos, como no caso das perguntas. A Figura 20 demonstra a representação em XML destes mesmos atributos.

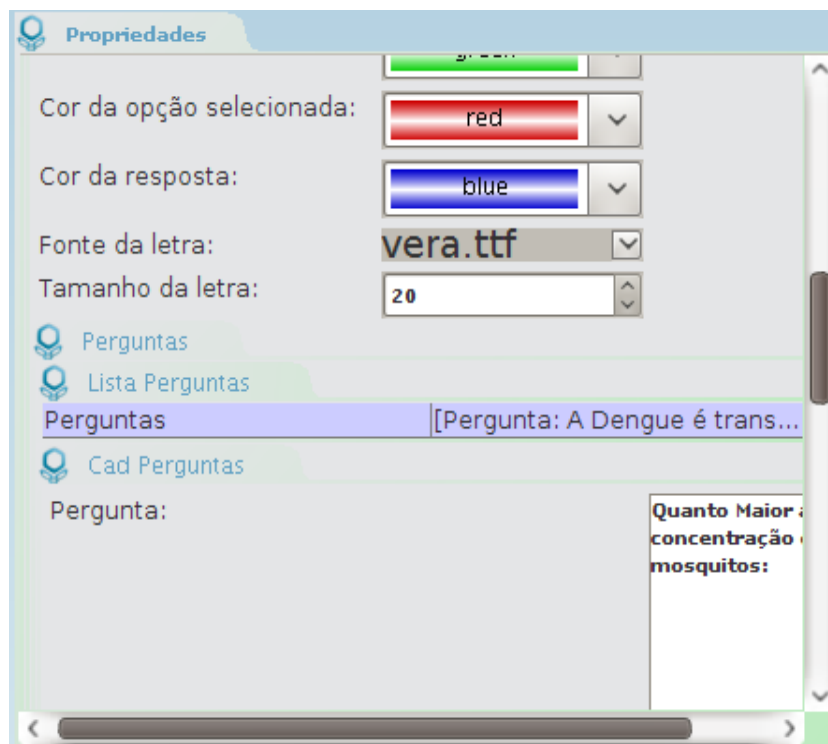


Figura 19: Painel de propriedades do "Jogo de Perguntas"

```

<item nome="Cor da opção selecionada" type="corOpcao" tipo="cor" default="red"/>
<item nome="Cor da resposta" type="corResultado" tipo="cor" default="white"/>
<item nome="Fonte da letra" type="fonte" tipo="fonte" default="vera.ttf"/>
<item nome="Tamanho da letra" type="fontsize" tipo="numero" default="20"/>
<itemGrupo nome="Perguntas" type="perguntas" min="1">
  <item nome="Pergunta" type="pergunta" default="Pergunta?" tipo="textarea"/>

```

Figura 20: Representação em XML dos atributos do painel de propriedades da figura 19.

6.5 Geração de Programas com aplicativos Interativos no padrão Ginga-NCL.

No Célula um gerador é criado estendendo-se a interface “Gerador”, que possui apenas a declaração da função “gerar”. A função “gerar” recebe como parâmetro um objeto da classe “Projeto” e o diretório destino do programa. No caso do gerador para o padrão Ginga-NCL, primeiro todas as mídias que foram incluídas no projeto são copiadas para a pasta destino e depois o documento NCL é feito. O documento NCL é produzido através de templates pré-definidos para cada tipo de mídia, o template só substitui o nome do arquivo, da mídia e a cor do botão que lança a mídia.

Para os aplicativos, além de declarar e incluir o lançador no documento NCL, também é preciso passar os parâmetros modificados pelo usuário da ferramenta. No protótipo inicial da ferramenta a passagem de parâmetros para as interatividades em NCLua era feita através de arquivos de texto. Cada parâmetro era encapsulado em um arquivo, onde o nome do arquivo era o nome do parâmetro e o conteúdo era o valor do parâmetro. Desta forma o número de arquivos aumentava conforme o número de parâmetros. O número alto de arquivos poderia causar uma lentidão desnecessária, pois o script lua precisava abrir um arquivo de cada vez, armazenando os valores para futura utilização. Além da lentidão, o número variável de parâmetros em alguns programas acarretava um número variável de arquivos, podendo dificultar na identificação dos parâmetros. Um exemplo de programa com número de parâmetros indefinidos é o jogo de perguntas, onde tanto o número de perguntas quanto o número de alternativas para cada pergunta pode variar.

Para este problema, a proposta foi passar os parâmetros através de um arquivo XML, substituindo todos os arquivos de parâmetros por apenas um. O arquivo possui os elementos *group* e *object*, ambos possuem o atributo *type* que recebe o nome do atributo em questão, definido no arquivo de definições do modelo. Além do atributo *type*, outras propriedades do parâmetro podem ser passadas através do arquivo, que serão passadas através de atributos do elemento no documento XML. Um exemplo seria um texto (Figura 21), que pode ter várias propriedades, como a fonte, o tamanho da fonte, a cor e entre outros.

```
<object color="black" font="vera.ttf" fontsize="20" position="center,30%,86%,68%"  
type="texto">Texto Exemplo</object>
```

Figura 21: Elemento "object" do arquivo de parâmetros

O elemento *group* é utilizado para separar os parâmetros que se repetem dos outros, como no exemplo do jogo de perguntas (Figura 22), neste caso, haveria um grupo para as perguntas, e para cada pergunta, um grupo para as alternativas.

```
<group type="perguntas">  
  <object type="pergunta">Pergunta 1?</object>  
  <group type="alternativas">  
    <object pontuacao="0" type="alternativa">Alternativa 1</object>  
    <object pontuacao="1" type="alternativa">Alternativa 2</object>  
    <object pontuacao="0" type="alternativa">Alternativa 3</object>  
    <object pontuacao="0" type="alternativa">Alternativa 4</object>  
  </group>  
  <object type="pergunta">Pergunta 2?</object>  
  <group type="alternativas">  
    <object pontuacao="1" type="alternativa">Alternativa 1</object>  
    <object pontuacao="2" type="alternativa">Alternativa 2</object>  
    <object pontuacao="3" type="alternativa">Alternativa 3</object>  
  </group>  
</group>
```

Figura 22: Grupo de parâmetros

Desta forma ficou especificada a maneira como a ferramenta passa os parâmetros para o programa, utilizando apenas um arquivo. A função que gera o projeto precisa, para cada aplicativo, gerar também o documento XML de parâmetros, que tem como padrão o nome “objects.xml”. Para cada aplicativo escrito na linguagem Lua, é adicionada uma linha no começo do arquivo Lua (Figura 23), criando um objeto chamado “objects” apontando para o documento de parâmetros. Assim o criador do modelo não declara esta variável, mas extrai todos os parâmetros desta variável.

```
local objects = "_objects.xml"
```

Figura 23: Linha adicionada no arquivo Lua pelo Célula

6.6 Internacionalização do Célula

Como o Célula é desenvolvido em Java, a internacionalização foi feita usando o padrão definido pelo Java para internacionalização. Em softwares desenvolvidos na linguagem Java, todo software tem um objeto da classe *Locale* que indica a língua e a localidade do software. Por padrão, este objeto é criado utilizando a língua determinada para a máquina virtual instalada. Para que o software carregue as *strings* correspondentes à língua determinada pelo *Locale*, é criado um arquivo para cada língua que guarda as *strings* traduzidas para cada língua e um objeto da classe *org.jdesktop.application.ResourceMap* se encarrega de abrir o arquivo correspondente e carrega as *strings*. No Célula a internacionalização é feita desta forma. No protótipo inicial não havia nenhum suporte a internacionalização.

Para que o usuário do Célula tenha a liberdade de modificar a língua padrão, foi criado um botão no menu superior que ao ser clicado, uma nova janela é aberta e é possível escolher uma entre as línguas disponíveis. Ao selecionar uma das línguas, o usuário é informado que é necessário reiniciar o software para que a nova língua seja aplicada às *strings* do software. Atualmente o Célula tem suporte a apenas duas línguas, inglês e português brasileiro. A Figura 24 mostra o botão e a janela que é aberta para mudar a língua padrão do software.



Figura 24: Internacionalização no Célula

A internacionalização do Célula, internamente é feita com um documento XML no diretório raiz, chamado “configuration.xml”. Neste arquivo são armazenadas configurações básicas do software, entre elas o *Locale* escolhido pelo usuário. Outra parte do Célula que também contém *strings* que precisam ser traduzidas está nos modelos de interatividade. Para isto, no diretório dos modelos foi criada uma pasta para

cada língua disponível, assim, o desenvolvedor precisa criar um arquivo traduzido para cada língua disponível. O repositório de interatividades irá sempre abrir a pasta correspondente à língua definida ao abrir o programa.

7 Modelos de Interatividade Criados para a Ferramenta

A criação dos modelos de interatividade foi feita através de testes e adaptações. Era criado um exemplo do modelo, testado no ambiente de testes, o Virtual Ginga-NCL STB. Depois o modelo é adaptado para que a ferramenta gere o programa sem erros e de acordo com o padrão Ginga-NCL. Neste trabalho foram feitos quatro modelos, um jogo de memória, um jogo de perguntas e respostas (chamado de quis), uma galeria de imagens e um modelo para informação textual, sendo que este último pode ou não aceitar uma imagem.

O aplicativo pode receber parâmetros pela ferramenta, isto é feito através do documento XML de definição da interatividade, onde são definidos os atributos específicos. Estes atributos podem ser modificados pelo usuário através do *Célula* e são passados ao aplicativo através de um documento de texto, escrito em XML, onde são escritos todos os argumentos definidos anteriormente pelo documento de definições.

Lua é uma linguagem baseada em tabelas, e existem muitas facilidade para lidar com elas, como índices dinâmicos (aceita qualquer tipo de dado, com exceção de nulo) e funções de manipulação. A partir disto e com o objetivo de aumentar o reuso de código, foi usada uma biblioteca de leitura de XML, que faz o parsing das tags e atributos para uma tabela Lua, com a facilidade de poder manipulá-la.

As imagens a seguir são feitas a partir do emulador Ginga-NCL Virtual STB, testando aplicativos interativos gerados pelo *Célula* a partir dos modelos de interatividade criados.

7.1 Jogo de Memória

O jogo de memória (Figura 25), por se tratar de um jogo, foi o único que precisou de uma lógica um pouco diferente dos demais modelos. Possui treze atributos:

- uma imagem de fundo, que tem como propriedade a posição, para ser alterada;
- uma imagem para a carta virada e outra para a carta selecionada;
- oito imagens para o jogo, o que dá 8 pares em um tabuleiro de 4 por 4;

- e ainda um atributo “servlet”, que é o caminho absoluto de um servidor de aplicação que armazena as estatísticas dos jogadores.



Figura 25: Aplicativo gerado pela ferramenta a partir do modelo “Jogo de Memória”

7.2 Informação Textual

Um modelo para informação textual (Figura 26) já havia sido feito no protótipo inicial do Célula, porém, era totalmente diferente da versão atual. Na primeira versão ele só continha o texto como parâmetro e a imagem era fixada pelo criador do modelo, além de não ser necessária a leitura do documento XML. Uma das dificuldades encontradas foi a ideia de incluir uma barra de rolagem, que é incluída quando há um texto muito grande para ser exibido.

Este modelo aceita duas maneiras de ser exibido, uma sem e outra com uma imagem extra, além da imagem de fundo, no caso de se querer apresentar uma informação sobre a imagem em questão. Portanto o modelo possui dois ou três atributos:

- uma imagem de fundo, com a posição como propriedade para ser modificada;
- o texto, que é do tipo “área de texto”, e possui quatro propriedades, a posição, a fonte, cor e tamanho da letra;
- uma imagem, que pode ou não ser incluída no documento a ser carregado pela ferramenta, que possui, assim como os outros, a posição como uma propriedade.

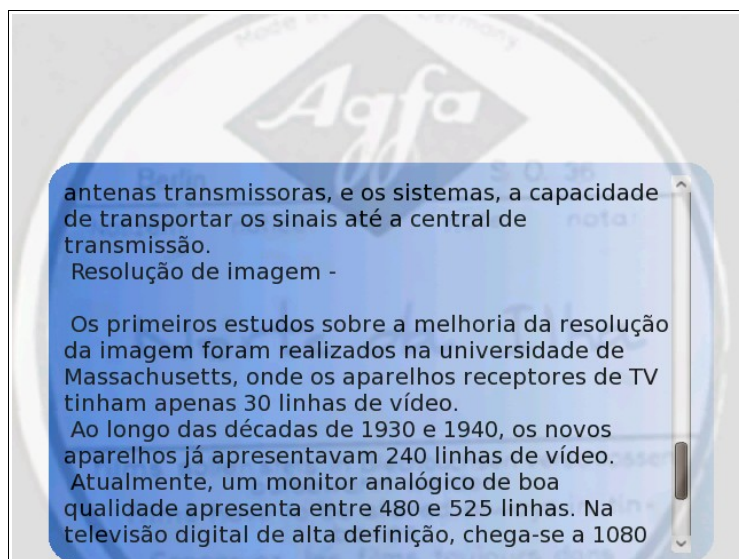


Figura 26: Aplicativo gerado pela ferramenta a partir do modelo “Informação Textual”

7.3 Jogo de Perguntas

O jogo de perguntas (Figura 27) também já havia sido criado na primeira versão do software, mas com algumas diferenças. Na primeira versão ainda não tinha o conceito de grupo de atributos, portanto, o jogo só tinha uma pergunta e o número fixo de 5 alternativas. Desta forma, a maior mudança de uma versão para outra foi a introdução do número variável de perguntas, alternativas e respostas.

Este foi o modelo com maior complexidade para tratar da leitura do arquivo XML de argumentos, isto porque ele possui o número de perguntas e respostas variado, ou seja, o usuário, criador da interatividade é quem define a quantidade de perguntas e as regras associadas à pontuação adquirida nas perguntas. É o modelo com maior número de argumentos:

- uma imagem de fundo, com a possibilidade de alterar sua posição;
- um argumento para a posição das perguntas em relação à tela;
- a cor das alternativas, da pergunta, da opção selecionada e da resposta;
- a fonte e o tamanho da letra;
- um grupo de atributos para as perguntas, que tem uma área de texto para a pergunta e mais um grupo para as alternativas da pergunta, que possui o texto da alternativa e uma propriedade para a pontuação daquela alternativa;

- por último é adicionado um grupo de regras para as respostas do quiz, que possui o intervalo da pontuação como propriedades e o texto para a resposta.

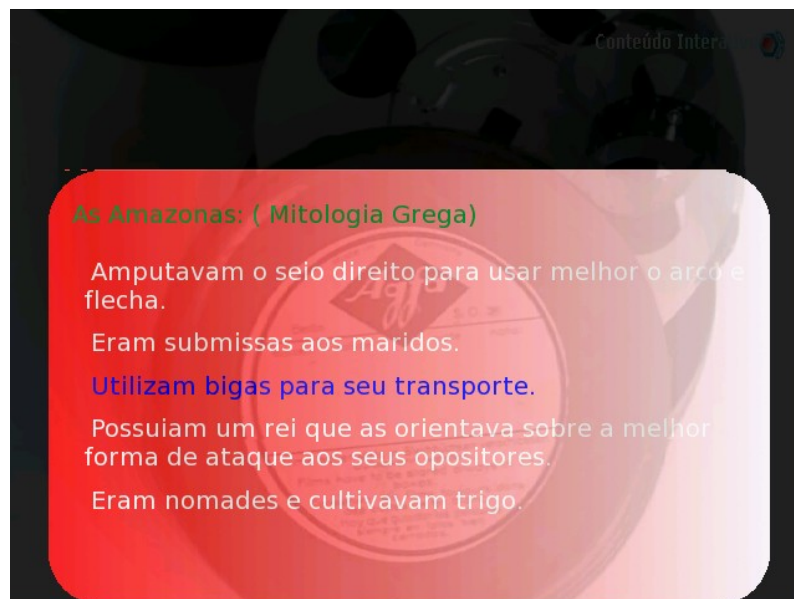


Figura 27: Aplicativo gerado pela ferramenta a partir do modelo “Jogo de Perguntas”

7.4 Leitor de Notícias em RSS

O leitor de notícias (Figura 28) utiliza um arquivo RSS, geralmente disponível online em páginas de notícias. O programa utiliza um módulo de acesso ao protocolo TCP, que faz esta comunicação através da classe *tcp* no módulo *event*, disponível no padrão Ginga-NCL. Assim que é retornado o arquivo de notícias, é criada uma tabela com todas as notícias disponíveis. Desta forma os argumentos são:

- o caminho para o documento de notícias RSS;
- um inteiro que determina o intervalo, em segundos, entre cada notícia;
- uma imagem de fundo com a posição como propriedade;
- duas outras imagens para indicar que clicando para esquerda ou para direita é possível avançar ou retornar para outras notícias;
- mais três outros argumentos para definir a cor, a fonte e o tamanho do texto.



Figura 28: Aplicativo gerado pela ferramenta a partir do modelo “Leitor de Notícias”

7.5 Galeria de Imagens

A ideia deste modelo foi fazer uma galeria de imagens, permitindo ao usuário percorrer as imagens através dos botões direcionais do controle remoto. Para o autor do aplicativo, também é possível adicionar uma legenda para cada imagem. Este foi o último modelo de interatividade criado e ele possui os seguintes atributos:

- Cor, fonte e tamanho da letra utilizada nas legendas;
- Um grupo de atributos para as imagens, contendo um atributo para a imagem e outro para o texto, os dois com a região como parâmetro.

A Figura 29 mostra duas telas de um aplicativo gerado a partir do modelo “Galeria de Imagens”, que é composto com imagens de flores, no canto superior direito do vídeo.

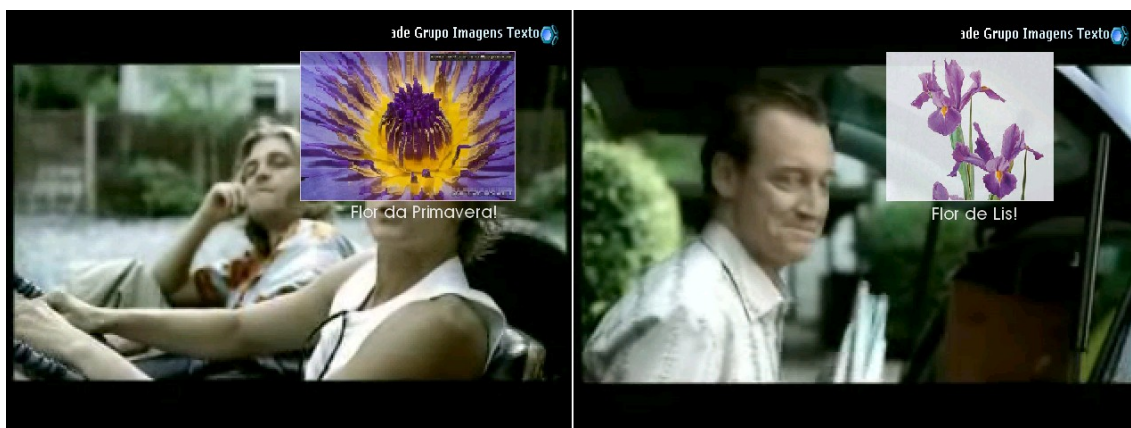


Figura 29: Telas de uma galeria de imagens criada a partir do modelo "Galeria de Imagens".

8 Conclusões

O principal objetivo deste trabalho de conclusão era propor melhorias para uma ferramenta de autoria para televisão digital, em questões de usabilidade e de funcionalidades. No começo do trabalho já havia um protótipo inicial dela, as propostas iriam modificar esta primeira versão da ferramenta.

A metodologia praticada para a execução do trabalho se incluiu uma pesquisa das funcionalidades e possibilidades permitidas com o padrão brasileiro de televisão digital. Também envolveu uma análise das ferramentas já existentes no mercado e na bibliografia, além de uma pesquisa das funcionalidades que um jornalista, o usuário final do software, espera de uma ferramenta de autoria.

As propostas foram feitas e muitas melhorias foram implementadas com o objetivo de comprovar se surtiram efeito para os usuários finais. Os aplicativos interativos desenvolvidos funcionaram de acordo com o padrão brasileiro Ginga-NCL e foram criados a partir da ferramenta *Célula*. O método de criação de modelos de interatividade para a ferramenta de autoria *Célula* se mostrou eficiente e os três modelos criados foram testados por profissionais de jornalismo, que aprovaram as interatividades. A exceção para a inclusão dos aplicativos interativos são para aqueles que fazem uma integração maior com o documento NCL, pois assim a ferramenta precisaria de modificações na hora da geração do NCL.

8.1 Trabalhos futuros

Muitas funcionalidades podem ser incluídas na ferramenta. Como a adição de novos modelos de interatividade, a integração da ferramenta com um software que automatize o envio de programas para uma emissora de televisão, tutoriais de uso da ferramenta, melhorar o painel de pré-visualização para executar o vídeo completo com as interatividades incluídas, ou seja, um simulador do programa, entre outros.

Os modelos de interatividade desenvolvidos no padrão Ginga-NCL são apenas testes, e qualquer outro aplicativo interativo pode ser incluído na ferramenta a partir do método de inclusão proposto. É possível imaginar muitos novos aplicativos, como jogos educacionais, jogo de perguntas com imagens, aplicativos com canal de retorno, além

da melhoria dos modelos já criados, a partir da adição de novos argumentos, por exemplo.

Ainda pensando nos modelos de interatividade, é possível pensar em outros tipos de atributos, como “booleano”, que pode assumir apenas dois valores, “float”, ou ponto flutuante para números no domínio dos reais. Mudanças no método de entrada dos atributos também podem ser melhorados, como o editor da região, que poderia ter opções para mudar a região sem perder a proporção e a opção para visualizar a região proporcionalmente ao tamanho do vídeo principal, pois a região mostrada é referente ao tamanho do aplicativo.

Referências Bibliográficas

Alticast, **Alticomposer**. Disponível em: <<http://www.alticast.com/main.html>> Acesso em: 20 out 2009

BECKER, Valdecir; MORAES, Áureo. **Do analógico ao Digital**: uma proposta de comercial para TV interativa. IN: III Simpósio Catarinense de Processamento Digital de Imagens. 2003. Florianópolis, Anais do Simpósio Catarinense de Processamento Digital de Imagens, 2003.

CASTRO, César A. S., GIUSEPPE, Loredana Di. **Como a mídia televisiva aborda os temas saúde do trabalho e meio ambiente**. 2007, p.2. In: WORKSHOP EM GESTÃO INTEGRADA: Riscos e Desafios. São Paulo, SP. 25 maio 2007. Disponível em <<http://www1.sp.senac.br/hotsites/cas/workshopgestaointegrada/files/03.pdf>>. Acesso em: 6 jun 2009.

CROCOMO, Fernando A. **TV Digital e produção interativa: a comunidade manda notícias**. Florianópolis: Editora da UFSC, 2007, p.40-70.

CROCOMO, F. A., Wangenheim, A. V., Comunello, E., Weber, M. H. ; Schneider, G. P., Munhoz, J. G. **Produção de Conteúdo para TV Digital e Uso de Ferramenta de Autoria para Inserção de Interatividade**. In: XXXII Congresso Brasileiro de Ciências da Comunicação, 2009, Curitiba - PR. Produção de Conteúdo para TV Digital e Uso de Ferramenta de Autoria para Inserção de Interatividade, 2009.

D'AVILA, Nícia. **Região Sudeste: Hegemonia na Mídia Televisiva**. 2007. Marília, PR: [s.e.] p.2, 2007. Disponível em <<http://www.scribd.com/doc/9636702/Regiao-Sudeste-Hegemonia-na-Midia-Televisiva-Nicia-DAvila>>. Acesso em: 6 jun 2009.

Fantástico. **Mudança Geral**. 2009. Disponível em <<http://fantastico.globo.com/Jornalismo/FANT/0,,MUL1104543-15605,00.html>>. Acesso em: 09 nov 2009.

FLORES, Luciano V; FAUST, Richard; PIMENTA, Marcelo S. **Definindo uma Proposta para Avaliações de Usabilidade de Aplicações para o Sistema Brasileira de TV Digital**. In: VIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC 2008), 2008, Porto Alegre. Anais do VIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais. Porto Alegre : SBC, 2008. p. 88-97.

GUIMARÃES, R. L. **Composer: um ambiente de autoria de documentos NCL para TV digital interativa**. Dissertação de Mestrado. Rio de Janeiro: PUC-RJ, 2007. 106 p.

Fraunhofer IAIS, **Jame Author**. Disponível em: <<http://www.iais.fraunhofer.de/jame.html?&L=1>>. Acesso em: 03 dez 2009.

Ginga NCL, **Ginga-NCL Virtual STB**. Disponível em: <<http://www.gingancn.org.br/ferramentas.html>>. Acesso em: 25 mar 2009.

Icareus, **Icareus iTV Suite Author**. Disponível em: <<http://www.icareus.com/web/guest/technologies/itvsuite/author>>. Acesso em: 3 nov 2009.

JAME Autor. Disponível em: <<http://www.iais.fraunhofer.de/jame.html?&L=1>>. Acesso em: 13 out 2009.

Java, Linguagem de Programação. Disponível em <<http://java.sun.com/>>. Acesso em: 11 mar 2009

WEBER, M. H., Bastos, R. C., Comunello, E., Crocomo, F. A., Krieger, J., Wangenheim, A. V. **Método para produção de interatividade educativa na TV Digital**. In: XX Simpósio Brasileiro de Informática na Educação, 2009, Florianópolis. anais do XX Simpósio Brasileiro de Informática na Educação, 2009.

MELONI, Luis G. **Return Channel for Brazilian Digital Television System-Terrestrial**, [JBCS] Journal of the Brazilian Computer Society, n.1; Vol. 12; Mar. 2007

MONTEIRO, Renata. **A Violência no Brasil: Como a mídia televisiva atua?**. 2009, p.1. In: Reportagem no site Blog da Comunicação. Disponível em <<http://www.blogdacomunicacao.com.br/?p=4507>>. Acesso em: 6 jun 2009.

MONTEZ, Carlos, BECKER, Valdecir. 2005. **TV Digital Interativa: Conceitos, Desafios e Perspectivas para o Brasil**. 2ed. Florianópolis: Ed. da UFSC.

MONTEZ, Carlos; BECKER, Valdecir. **TV Digital Interativa: Conceitos e Tecnologias**. In: WebMidia e LA-Web 2004 - Joint Conference. Ribeirão Preto, SP, 2004.

MORENO, Marcio F. **Um middleware declarativo para sistemas de TV digital interativa**. 2006. 77f. Dissertação (Mestrado em Informativa) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

PALHA, Cassia R.L. **Mídia televisiva e história: apontamentos metodológicos**. 2006. In: III Seminário Nacional de História Cultural. Florianópolis, SC. p. 2, ano 12, n. 9, dez 2006.

RODRIGUES, Rogério Ferreira; SOARES, Luiz Fernando Gomes. **Produção de Conteúdo Declarativo para TV Digital**. In: Anais do Seminário Integrado de Software e Hardware. 2006. Campo Grande, XXXIII SEMISH – Seminário Integrado de Software e Hardware. Porto Alegre: Sociedade Brasileira de Computação, 2006.

SANTOS, Adriana Cristina Omena. **Reflexões sobre a convergência tecnológica: A TV. Digital interativa no Brasil**. 2003. Disponível em: <http://bocc.ubi.pt/pag/santos-adriana-tv-digital-interactiva-no-brasil.html>. Acesso em: 24 nov 2009

SILUGEM, Dirce M.; DEVINVENZI, Macarena U.; LESSA, Angelina C. Diagnóstico do Estado Nutricional da Criança e do Adolescente. J Pediatr (Rio J). 76 Supl. 3, S274-S284, dez 2000.

SILVEIRA, Sérgio Amadeu. **Exclusão digital: a miséria na era da informação**. São Paulo: Perseu Abramo, 2001.

SIMÕES, Inimá. **A nossa TV Brasileira: por um controle social da televisão**. São Paulo: Editora Senac, 2004.

SOARES Luiz F.; RODRIGUES, Rogério F.; MORENO, Márcio F. **Ginga-NCL: the Declarative Environment of the Brazilian Digital TV Sistem.** In: [JBCS] Journal of Brazilian Chemical Society, Revista n.1; Vol. 12; Mar. 2007.

SOUZA, David N. ; SILVA, Ricardo D. ; RODRIGUES, Rodrigo L. ; LULA, Waltimar B. R. **Inclusão Digital e Software Livre: Elementos Auxiliadores ao Combate a Exclusão Social** - Uma Abordagem Do Projeto Id@L/Uepb. In: XXV Congresso Nacional dos Estudantes de Computação - ENECOMP 2007, Cuiabá. XXV Congresso Nacional dos Estudantes de Computação - ENECOMP, 2007.

SOUZA JÚNIOR, P. J. LuaComp: uma ferramenta de autoria de aplicações para TV digital. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM 375/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 143 p.

STUMM, Jaqueline. **Sistema Inteligente de Monitoração Alimentar via Web Baseado em Lógica Fuzzy.** In: Dissertação (Mestrado) Curso de Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis.

SUN MICROSYSTEMS, **Java Technology.** Disponível em <<http://java.sun.com>>. Acesso em 12 de março de 2009.

SUN MICROSYSTEMS, **Java Tutorial, Internacionalização.** Disponível em: <<http://java.sun.com/docs/books/tutorial/i18n/index.html>>. Acesso em: 5 fev 2010.

TAVARES, Walkyria M. L. **Televisão digital:** viabilidade de seu uso como instrumento de inclusão digital. Monografia Final de Curso de Especialização. Brasília: UnB, jun 2006. Disponível em: <<http://www.camara.gov.br/internet/infdoc/novoconteudo/acervo/temas/TVDigitalWalkiria.pdf>>. Acesso em: 25 nov 2009.

Tudo sobre TV: A História da televisão no Brasil – Anos 50. Disponível em <<http://www.tudosobretv.histor.tv/historbr.htm>>. Acesso em: 6 jun 2009.

VMWare, **VMWare Player**. Disponível em: <<http://www.vmware.com/>>. Acesso em: 25 mar 2009.