Universidade De São Paulo – USP

Instituto de Ciências Matemáticas e da Computação ICMC

Programa de Pós-Graduação



**Construção de uma interface gráfica para geração de arquivos NCL**

**SCC 5811 – Fundamentos de Sistemas Hipermídia e Web**

Matheus Ricardo Uihara Zingarelli - 5377855

Tiago Henrique Trojahn  - 7484581

São Carlos, SP.

19 de maio de 2010

# 1. Introdução

O sinal do Sistema Brasileiro de Televisão Digital (SBTVD) está se expandindo e já atinge a maioria das capitais e grandes cidades brasileiras. Com isso, pode-se esperar um crescimento no desenvolvimento de aplicativos para serem executados na TV Digital. Uma das maneiras mais utilizadas para desenvolvimento é fazendo uso da dupla NCL-Lua.

Até o momento, o desenvolvimento de aplicações NCL, é feita de duas formas:

(1) codificando diretamente o arquivo ou (2) através do auxílio de uma ferramenta GUI chamada *Composer*. A codificação direta em arquivo baseia-se na escrita do código fonte utilizando-se um *plugin* para a IDE Eclipse. Tal metodologia necessita que o criador de aplicações tenha que criar o programa e testá-lo a cada modificação na máquina virtual, causando demoras no desenvolvimento, fora é claro a necessidade de possuir conhecimento da linguagem NCL. Para resolver, ou pelo menos amenizar tal problema, é que foi criado o aplicativo chamado *Composer*, tendo por objetivo prover uma visão gráfica de alto nível do aplicativo em desenvolvimento. Problemas de incompatibilidade, erros no aplicativo, tanto no funcionamento como na instalação e diversas outras limitações levaram o *Composer* a ser descontinuado.

Este projeto tem o objetivo de criar uma aplicação online chamada de **NCLCreator** capaz de facilitar a criação de alguns dos recursos visuais mais básicos de um programa NCL: a definição e os detalhes de posicionamento e tamanho de figuras, vídeos e regiões do NCL. O usuário iria manipular visualmente estes elementos e a aplicação geraria o código NCL inicial. Por ser online, não seria necessária a instalação de nenhum software adicional específico, bastando o usuário possuir um navegador atual que permitisse execução de javascript. Por ser uma interface gráfica, permitiria a usuários não especialistas a criação de aplicativos simples em NCL sem conhecimento da linguagem.

# **2. Ferramentas utilizadas**

As tecnologias que foram utilizadas são aquelas com a qual o grupo já possui certa experiência, facilitando a implementação. Segue abaixo uma pequena descrição de cada uma delas. Na seção 6 pode ser encontrada uma lista de materiais de referência e tutoriais que utilizamos no desenvolvimento, separados por cada tecnologia.

XML: é uma padronização da W3C (*World Wide Web Consortium*) para linguagens de marcação. Criada para facilitar a comunicação entre computadores e também a leitura por seres humanos. É extensível, permitindo ao programador a criação e definição de tags de forma simples e rápida. É também a base de diversas linguagens de programação baseadas em tags, tais como o NCL.

NCL (*Nested Context Language*): é uma linguagem declarativa baseada em XML voltada para o sincronismo espaço-temporal de mídias para a TV Digital Brasileira. O objetivo do projeto é a criação automática de um arquivo NCL de um aplicativo cujo design foi feito utilizando a interface gráfica disponibilizada pelo NCLCreator.

PHP: é uma linguagem de programação voltada ao servidor, criada para facilitar o desenvolvimento de websites dinâmicos. Possui recursos poderosos tais como orientação a objetos, métodos e funções, ainda mantendo simplicidade na linguagem, tal como a tipagem dinâmica. Além disso, ele possui vasta documentação e comunidade de usuários. No projeto, o PHP foi utilizado para criação do arquivo NCL resultante da interação do usuário com a aplicação, bem como apresentação da interface da aplicação, junto com o HTML e CSS.

jQuery: é uma biblioteca javascript criada para promover várias funções úteis e reutilizáveis ao programador, agilizando o processo de desenvolvimento. Além disso, é padronizada para ser executada em diferentes navegadores de forma compatível. Também possui documentação extensa e grande comunidade de usuários. Seu sucesso é tamanho que gerou uma grande quantidade de projetos menores voltados para partes específicas do desenvolvimento Web. No projeto, o jQuery serve como uma ponte entre os dados gerados na interação do usuário e o PHP.

jQuery UI: é uma das bibliotecas resultantes dos mencionados projetos menores derivados do jQuery. Ela provê uma série de recursos visuais para interação do usuário com uma aplicação Web, tais como *widgets*, temas e animações. Foram utilizados no projeto os *plugins* que simulam a interação de um usuário com uma aplicação *desktop*, habilitando as funções de selecionar, arrastar e redimensionar elementos apresentados na interface sendo executada no navegador Web.

Máquina Virtual GingaNCL: O programa NCL criado pelo NCLCreator foi testado na máquina virtual criada pela PUC-Rio com o objetivo de desenvolvimento de aplicativos NCL. No projeto, a máquina virtual foi utilizada com fins de validação e testes dos aplicativos exportados pelo NCLCreator.

# **3. Justificativas**

O aplicativo como um todo foi desenvolvido para ser hospedado em um sistema web para facilitar o desenvolvimento de uma aplicação simples, por parte do usuário, sem necessitar de qualquer software específico e/ou incompatibilidade com futuros *releases*. Para acessar o NCLCreator, basta um navegador moderno que execute javascript e uma conexão com a internet.

A interface foi planejada para ser simples, com cada opção descrevendo a função a que serve. Ela é dividida em duas regiões: a superior é a “área de trabalho” do usuário, na qual ele irá inserir os elementos que formam o aplicativo NCL, que poderão ser arrastados e redimensionados; a inferior é a área de ferramentas, na qual ficam disponíveis as funções de criação de elementos do aplicativo e por onde o usuário poderá baixar o arquivo NCL criado. O arquivo é armazenado no servidor assim que o usuário aperta o botão “Gerar NCL”. Como fins de simplicidade, o arquivo é reescrito toda vez que o usuário aperta o botão, não sendo possível recuperação após a reescrita.

Para se gerar o arquivo NCL, tornou-se necessário criar uma classe em PHP chamada de NCLDocument, responsável por receber as informações dos elementos passados via javascript e onde são especificados os métodos de inserção de dados como região e media, além do método responsável por criar a estrutura correta do XML.

# 4. Resultados obtidos

O aplicativo implementado é capaz de gerar regiões, inserir imagens e vídeos, os quais podem ser redimensionados e movidos pela área de trabalho. A tela inicial do aplicativo é apresentada na Figura 1. Na Figura 2, vemos um exemplo de como é a entrada de um arquivo de mídia (vídeo ou imagem). Um exemplo do aplicativo já desenvolvido e com a opção de baixar o arquivo exportado já disponível é apresentado na Figura 3. O arquivo NCL gerado foi testado através de sua execução na máquina virtual do GingaNCL é apresentado na Figura 4.

A demo do aplicativo está disponível para testes em: <http://www.icmc.usp.br/~zinga/NCLCreator/>

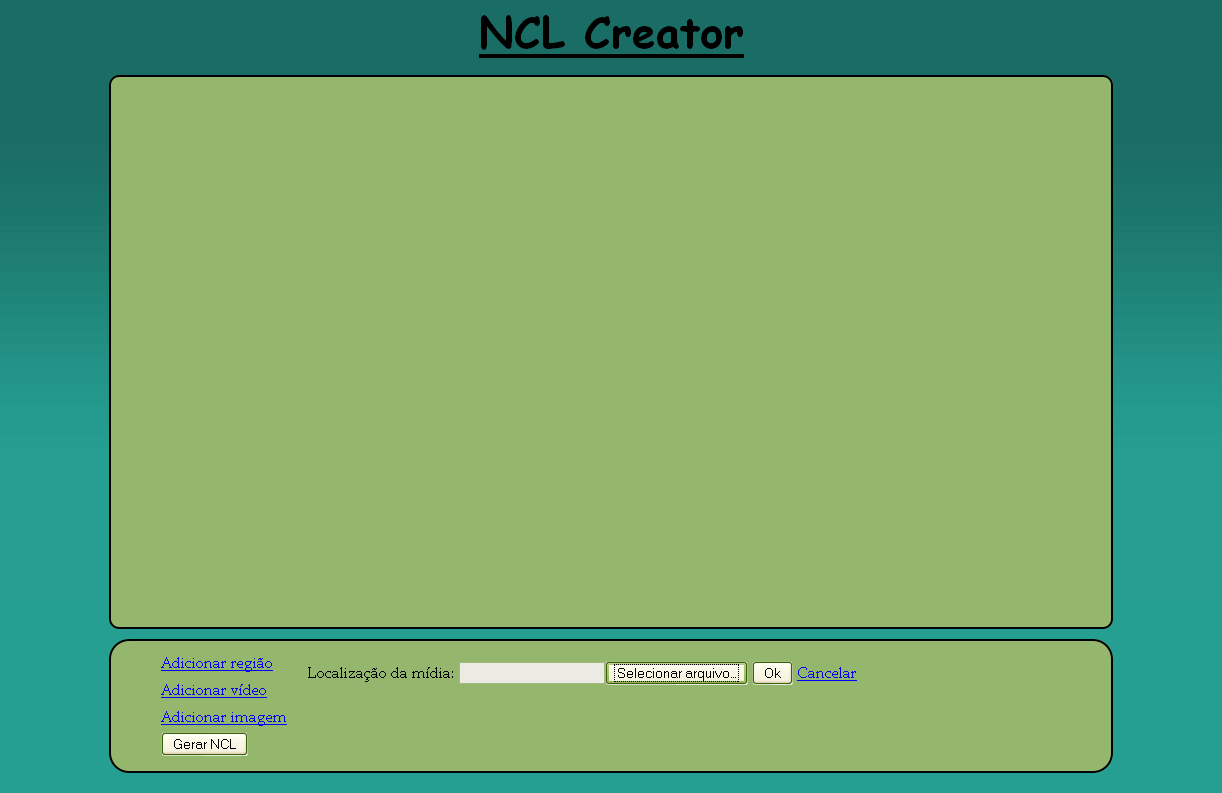


Figura 1 - Página inicial da aplicação NCLCreator.

Figura 2 – Submenu de adição de mídia.

# 5. Dificuldades encontradas e trabalhos futuros

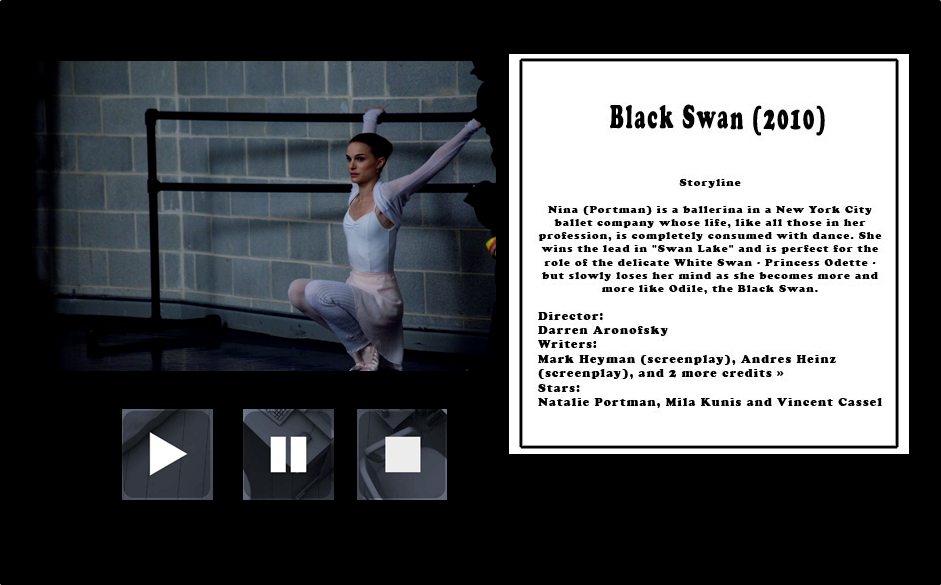
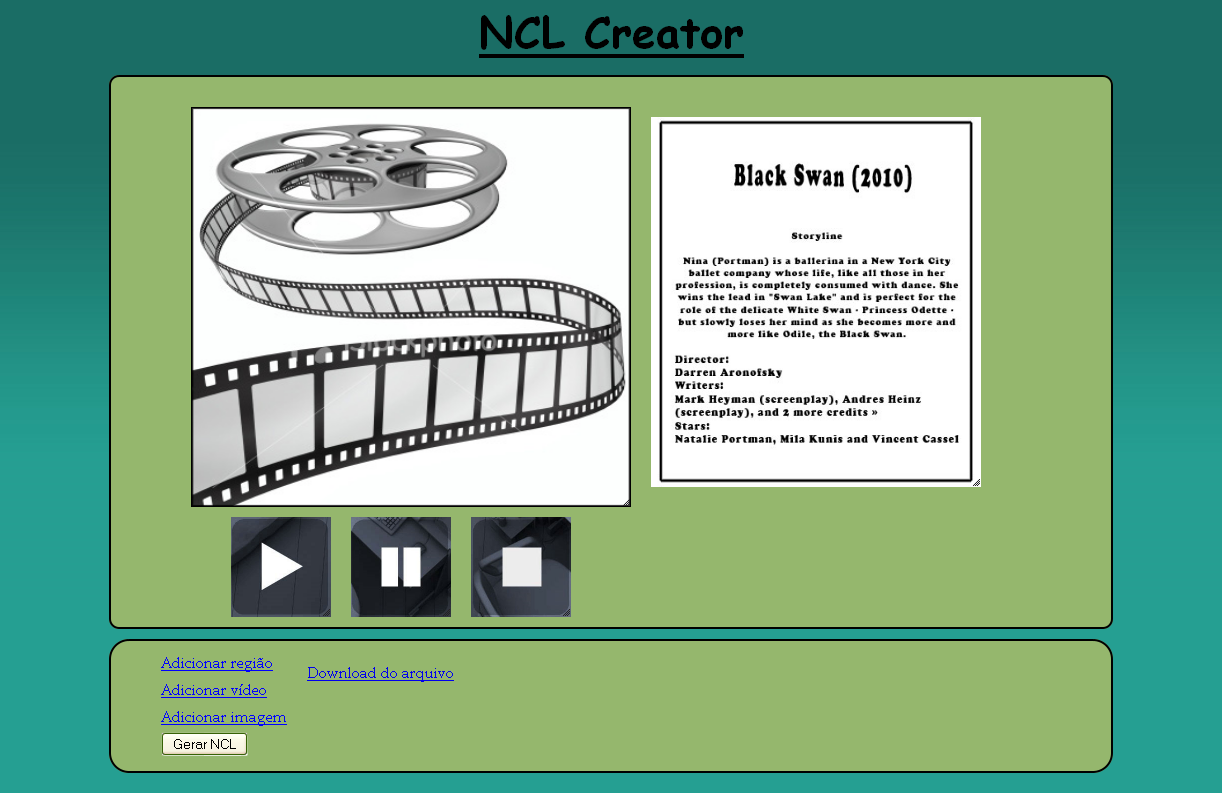


Figura 4 – Execução em uma máquina virtual do arquivo NCL gerado pela aplicação.

Figura 3 – Exemplo de um aplicativo completo, com adição de vídeo e imagens.



Uma das maiores dificuldades técnicas foi a de fazer a comunicação do javascript com o PHP para repassar as informações dos elementos. Uma das funcionalidades que gostaríamos de ter adicionado era a possibilidade de o usuário selecionar uma mídia do seu computador, que seria então feito upload ao servido. Infelizmente, o envio via javascript para o PHP se mostrou complexo demais para o tempo de entrega do projeto. Neste caso, portanto, o usuário ao adicionar um vídeo ou uma imagem, possui a opção de selecioná-lo de seu computador, porém, não poderá visualizá-lo na aplicação. Com isso, é necessário ao usuário deixar os arquivos de mídia na mesma pasta em que se encontra o código rodando o NCLCreator.

Como trabalhos futuros destacam-se o sistema de upload supracitado e criação de regiões aninhadas. Há também a possibilidade de se adicionar todo o conjunto de opções relativas à interação, tais como as diretivas *link* e *conector,* já implementadas na classe NCLDocument. Com os dois sistema implementado seria possível criar aplicações complexas e aptas com a interatividade prevista para a TV Digital Brasileira. Isso, junto com a melhoria das diretivas já implementadas, como o índice de profundidade (z-index) e o sistema de regiões pai-filho, tornaria o NCLCreator apto a se realizar a maioria das funcionalidades presentes no aplicativo Composer, com a vantagem de não haver problemas de compatibilidade ou instalação de componentes. Mais duas opções que seriam interessantes de serem futuramente implementadas é poder desfazer uma ou mais ações anteriores (Ctrl+Z) e apagar elementos.

# 6. Materiais de referência

Disponibilizamos abaixo os links que nos possibilitaram o desenvolvimento da aplicação.

* NCL
  + Exemplos de aplicativos escritos em NCL: <http://www.ncl.org.br/exemplos/index_30.html>
* PHP
  + Manipulação da árvore DOM e criação de XML: <http://www.php.net/manual/en/class.domdocument.php>
  + Orientação a Objetos: <http://php.net/manual/en/language.oop5.php>
* jQuery / jQuery UI
  + API do jQuery: <http://api.jquery.com/>
  + jQueryUI – Draggable (possibilita arrastar elementos): <http://jqueryui.com/demos/draggable/>
  + jQueryUI – Selectable (seleção de elementos): <http://jqueryui.com/demos/selectable/>
  + jQueryUI – Resizable (permite redimensionar elementos): <http://jqueryui.com/demos/resizable/>
* Máquina Virtual Ginga NCL
  + Descrição das ferramentas: <http://www.gingancl.org.br/ferramentas.html>
  + Download da Máquina Virtual: <http://www.ncl.org.br/ferramentas/ubuntu-server10.10-ginga-i386.zip>

# Anexo

Na Figura 5 encontra-se o arquivo NCL gerado pela aplicação esboçada na Figura 3 e cujo resultado pode ser visualizado na Figura 4.

Figura 5 – Documento NCL gerado pela aplicação

