Uma Arquitetura para Integração TV – Internet e Enriquecimento de Conteúdo em Aplicações de TVDi

Thiago S. Teixeira, Manoel C. Silva Filho

Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO) – Campus Palmas – Caixa Postal 77.021-090 – Palmas – TO – Brasil

thiago.txr@gmail.com, mcampos@ifto.edu.br

Abstract. *With the possibility of Internet access by the Interactive Digital TV, a range of services/resources, already available on the web, are converging to the DTV world. In this work, it is proposed an architecture for supporting the development and execution of iDTV applications (NCLua applications) capable of using several of these services/resources. Two case studies were performed to evaluate the proposed architecture: a t-government application and an application capable of enriching television content.*

Resumo. *Com a possibilidade de acesso à Internet pela TV Digital Interativa, uma gama de serviços/recursos, já disponíveis na web, estão convergindo para o mundo de TVD. Neste trabalho, é proposta uma arquitetura para apoiar o desenvolvimento e execução de aplicações de TVDi (aplicações NCLua) capazes de utilizar vários destes serviços/recursos. Foram realizados dois estudos de caso para avaliar a arquitetura proposta: uma aplicação t-government e uma aplicação capaz de enriquecer conteúdos televisivos.*

1. Introdução

Vários padrões de Televisão Digital (TVD) têm se estabelecido e evoluído, de forma a beneficiar a sociedade com os seus novos recursos e progressos, em comparação com a TV Analógica. Um dos grandes benefícios que a TVD tem proporcionado é a interatividade, capaz de transformar o telespectador, até então passivo, que apenas recebia as informações transmitidas pelas emissoras de TV, em um usuário ativo [Herrero, Cesar e Vuorimaa 2003], com a faculdade de interagir com aplicações executadas diretamente no conversor de sinal digital (*Set-Top Box* ou STB).

Com o advento da interatividade na Televisão Digital (TVDi), não só áudio e vídeo podem ser entregues ao telespectador, mas também dados e aplicativos, os quais são executados pelo STB. No contexto de aplicativos para a TVDi, a *Internet* tem ganhado bastante espaço, uma vez que a arquitetura da TV Digital propicia comunicações bilaterais pelo canal de interatividade. A integração TVDi – *Internet* abre portas para a convergência de uma série de serviços e recursos, já disponíveis na *web*, para a TV Digital. Com isso, é possível estender conteúdos televisivos através de aplicações interativas que extraiam informações da *Internet*, com a finalidade de enriquecer o programa/assunto, em exibição na TV, com conteúdo proveniente da *web*.

No Brasil, a televisão é um item presente em quase todos os lares, o que torna a união TVDi – *Internet* favorável à inclusão digital, apesar da porcentagem de lares com acesso à *Internet* ainda ser baixa no país. Segundo a Pesquisa Sobre Uso das Tecnologias de Informação e Comunicação do Brasil – TIC Domicílios e Empresas 2011 [CETIC.Br 2011], apenas 38% dos brasileiros possuem acesso à Internet em casa.

Por outro lado, o Programa Nacional de Banda Larga do Governo Federal [MC 2013] está tentando mudar esta situação, buscando universalizar o acesso à *Internet* rápida no território brasileiro.

Adicionalmente, em conformidade com o Decreto nº 4.901, de 26 de novembro de 2003, que instituiu o Sistema Brasileiro de Televisão Digital (SBTVD), o Governo Federal almeja promover a inclusão social e digital no país através da TV. Este fato contribui para a disponibilização de serviços governamentais (*t-government*) na TVD por meio de aplicações interativas, o que inclui serviços que dependem da *Internet*.

O presente trabalho propõe uma arquitetura para aplicações de TVDi que suporte vários cenários de integração com serviços e recursos *web*, alguns dos quais podem enriquecer conteúdos televisivos. Para tanto, foram desenvolvidas duas APIs (*Application Programming Interface*) para auxiliar a construção de aplicativos para o Ginga-NCL (subsistema do *middleware* do padrão brasileiro de TV Digital, o Ginga [ABNT 2008]), capazes de se integrar ao motor de busca do Google (por meio do serviço *Google Custom Search*[1]) e realizar a leitura de fontes de dados RSS (*Really Simple Syndication*) na *web*. O *Google Custom Search* é um serviço disponibilizado pelo Google que possibilita a personalização do motor de buscas e a sua integração com outras aplicações.

Além das APIs desenvolvidas, a arquitetura proposta compreende outros componentes já disponíveis à comunidade de desenvolvimento. Para avaliação da eficiência desta arquitetura, foram realizados dois estudos de caso (aplicações de TVDi que acessam a *Internet*). O primeiro possibilita a utilização de serviços governamentais, por meio de RSS e *Web Service*, disponibilizados pela Secretaria da Segurança Pública do Tocantins (SSP-TO), enquanto que o segundo permite a extração e exibição de informações da *web* na tela da TV, utilizando o motor de buscas do Google e RSS.

As próximas seções deste artigo estão organizadas da seguinte forma: a Seção 2 acrescenta detalhes sobre a integração TVDi – *Internet*; a Seção 3 relata acerca de trabalhos publicados, relacionados com o presente trabalho; na Seção 4 são apresentadas tecnologias e paradigma de programação utilizados; a Seção 5 apresenta a arquitetura proposta; a Seção 6 apresenta os resultados alcançados com os estudos de caso desenvolvidos; ao fim, a Seção 7 expõe considerações finais e sugere trabalhos futuros.

2. TVDi – Internet

A televisão é um meio que, além de entregar entretenimento, age como um integrador social. De acordo com o IBGE, em sua Pesquisa Nacional por Amostra de Domicílios realizada em 2011, a TV está presente em 96,9% dos lares brasileiros [IBGE 2011]. Toda esta cobertura, aliada à *Internet*, pode ser uma poderosa ferramenta de inclusão digital no Brasil, levando acesso a uma grande parcela da população que ainda não está inserida na Era Digital. Esta aliança é possível através da TV Digital Interativa (TVDi).

Por meio da integração TVDi – *Internet*, aplicativos executados no STB podem fornecer, ao usuário de TV, experiências que vão além da programação exibida pelas emissoras, de forma a estender o seu conteúdo. Esta extensão pode ocorrer através de

[1] http://www.google.com/cse

recomendação de fontes de informações referente ao assunto em exibição, integração com sites como a *Wikipedia*[2], busca de vídeos e notícias relacionadas, entre outros.

Retornando ao contexto do acesso à *Internet* em si, para expandir o alcance da *Internet* no Brasil, em 2010 o Governo Federal lançou o Programa Nacional de Banda Larga (PNBL), criado a partir do Decreto nº 7.175, de 12 de maio de 2010, tendo como objetivo proporcionar *Internet* com banda larga à população brasileira, oferecendo baixo preço e boa qualidade, por meio de investimentos na ampliação da infraestrutura de telecomunicações e redução de impostos para terminais de acesso [MC 2013].

3. Trabalhos Relacionados

Alguns trabalhos que abordam o tema TVDi – *Internet* têm enfatizado sobre a forma como os conteúdos *web* são apresentados na tela da TV. Em [Lima Jr. et al. 2010], os autores apontam que usabilidade, navegabilidade, dimensões de tela e recursos computacionais devem ser levados em consideração, quando se trata de exibição de conteúdos *web* na televisão. Apoiando-se nesta preocupação, os autores construíram uma API de extensão ao *middleware Multimedia Home Platform* (MHP), do padrão europeu de TVD – *Digital Video Broadcasting* (DVB), para adaptação de conteúdo na TVDi, e uma aplicação de validação desta API que busca informações sobre jogos da Copa do Mundo da África (2010), disponibilizadas em um servidor *web*.

Em [Ferretti et. al 2007], um dos principais focos do trabalho também foi sobre a adaptação de conteúdo *web*, através da construção de uma arquitetura capaz de converter, automaticamente, documentos HTML advindos da *Internet*, resultando em uma apresentação apropriada na TVD. Este trabalho foi construído sobre a plataforma DVB – MHP, fazendo-se uso do subsistema DVB-J. Destaca-se a criação de vários *widgets* (mini aplicativos) como provas de conceito para avaliar a arquitetura desenvolvida, em específico um aplicativo visualizador de RSS, o qual exibe conteúdos de *feeds* RSS na televisão, obtidos através de *web sites*.

Já em [Ghisi et. al 2010], os autores defendem que no Brasil, após a implantação do padrão brasileiro de TVD, a produção de aplicações interativas para o Ginga ainda é reduzida e, em sua maioria, meramente informativa ao telespectador. O trabalho teve como tema a integração de aplicações interativas para TV Digital, no subsistema Ginga-NCL, com a *Web* 2.0. Foram construídas duas aplicações NCLua para a TVDi: uma aplicação integrada à rede social *Twitter*[3] e outra aplicação capaz de enriquecer o conteúdo televisivo através da integração com o site *Wikipedia*.

4. Materiais e Métodos

Esta seção apresenta tecnologias e paradigma de programação utilizados na construção da arquitetura proposta, APIs e estudos de caso desenvolvidos.

A arquitetura e implementações foram feitas para o *middleware* Ginga. O Ginga-NCL (subsistema do Ginga) foi o escolhido como ambiente declarativo para o desenvolvimento, testes e simulação. Assim, duas linguagens foram utilizadas: 1) *Nested Context Language* (NCL): linguagem declarativa baseada em XML para autoria de documentos hipermídia [NCL 2013], que dispõe de recursos para realizar

[2] http://www.wikipedia.org (site de enciclopédias no formato *Wiki*)
[3] http://www.twitter.com

sincronismo espaço-temporal, suporte a múltiplos-dispositivos e à produção ao vivo de aplicações interativas [Ginga-NCL 2013], entre outros; 2) Lua: linguagem de *scripts* para programação procedural, projetada para estender aplicações [Lua 2013]. O desenvolvimento com as linguagens NCL e Lua produzem aplicações NCLua.

Para representar este ambiente (*middleware* Ginga) foi empregado o Ginga-NCL Virtual STB (versão 0.12.4), máquina virtual contendo implementação de referência do Ginga, disponibilizada pelo Portal do Software Público Brasileiro[4].

A linguagem de programação Java foi empregada na construção das aplicações *web* utilizadas nos estudos de caso. Como *Integraded Development Environment* (IDE) foi utilizado o Eclipse[5], para desenvolver as APIs e os estudos de caso (aplicações NCLua e aplicações *web* em Java).

O paradigma de programação empregado foi a Programação Orientada a Objetos (POO), apesar da linguagem Lua não possuir o conceito de classe. Todavia, é possível utilizar o conceito de classes em Lua por meio de sua estrutura de dados `table`, e, assim, criar objetos a partir delas. Desta forma, os códigos ficam estruturados, mais limpos e de fácil manutenção e reutilização.

5. Arquitetura Proposta

A arquitetura proposta neste trabalho é composta por componentes distribuídos entre *Set-Top Box* e *gateway* (servidor *web* que serve de ponte para serviços disponíveis na *Internet*). Esta arquitetura tem como finalidade prover suporte a aplicações NCLua que possuem integração com serviços e/ou recursos *web*. A Figura 1 apresenta a arquitetura em questão, na qual as ligações indicam a possibilidade de comunicação e os itens coloridos em azul representam componentes desenvolvidos neste trabalho.

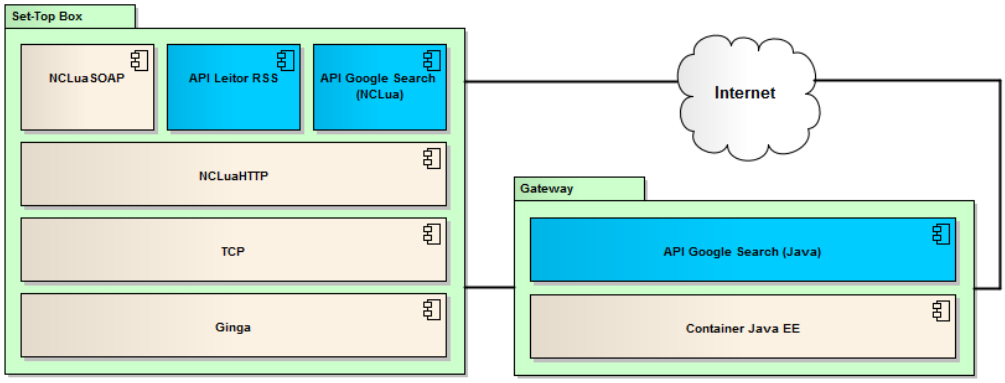

Figura 1. Arquitetura Proposta

Para a integração entre aplicações NCLua e a *Internet*, por meio do canal de interatividade, é necessário o uso de protocolos de comunicação padrões, como TCP, HTTP e SOAP [Silva Filho e Gondim 2011]. Para tanto, foram adicionados, à arquitetura proposta, os módulos TCP [Sant'anna 2013], NCLua_HTTP e NCLua_SOAP [Silva Filho e Gondim 2011].

O módulo NCLua_HTTP realiza a implementação de alguns dos recursos essenciais do protocolo HTTP/1.0 [Silva Filho e Gondim 2011]. Ele possui dependência

[4] http://www.softwarepublico.gov.br
[5] http://www.eclipse.org

do módulo TCP, representado na arquitetura. Já o módulo NCLua_SOAP realiza a implementação de recursos fundamentais do protocolo SOAP 1.1 e 1.2 [Silva Filho e Gondim 2011] e possui dependência com o módulo NCLua_HTTP. Ambos módulos foram escritos em Lua.

A API Leitor RSS (módulo em Lua) foi construída neste trabalho com o intuito de facilitar o desenvolvimento de aplicações NCLua para a TV Digital, que trabalham com a leitura de fontes de RSS disponíveis na *Internet*. Com a utilização do módulo NCLua_HTTP, esta API realiza uma requisição HTTP à *Internet*, buscando por um *feed* RSS. Para isso, a API informa o endereço completo para o arquivo na *web*, na requisição HTTP. Como o formato RSS é uma extensão do formato XML, o retorno da requisição se dá através de um arquivo em tal formato. Ao receber este arquivo, com o conteúdo do RSS, a API, através da utilização da biblioteca LuaXML [Silva Filho e Gondim 2011], converte o XML em uma tabela Lua e preenche sua estrutura própria de objetos, que representa o canal do RSS e as suas entradas. Ao concluir este processo, todo o conteúdo do RSS pode ser consumido por aplicações NCLua.

A API *Google Search* foi desenvolvida para permitir a integração de aplicações NCLua com o motor de buscas do Google, utilizando o serviço *Google Custom Search*. O seu propósito é dar suporte à realização de pesquisas de páginas *web* e sites através de aplicações NCLua. Ela possui uma arquitetura distribuída, na qual parte fica localizada no STB e parte em um *gateway*. A utilização do *gateway* se fez necessária, pois o serviço *Google Custom Search* exige uma requisição HTTPS para a realização de pesquisas, e o subsistema Ginga-NCL ainda não oferece suporte a tais requisições.

O seu funcionamento ocorre da seguinte forma: a API *Google Search* (NCLua) faz uma requisição HTTP, utilizando o módulo NCLua_HTTP, à outra parte da API presente no *gateway* (aplicação *web* Java executada em um servidor de aplicações Java *Enterprise Edition*). Ao receber a requisição, a API *Google Search* (Java) faz uma nova requisição, desta vez segura (HTTPS), ao serviço *Google Custom Search*, repassando os parâmetros de pesquisa enviados pela primeira requisição. O serviço *web Google Custom Search*, por sua vez, recebe a requisição, processa e retorna o resultado no formato *JavaScript Objetc Notation* (JSON), informando as páginas *web* encontradas em conformidade com as palavras-chave enviadas como parâmetro de pesquisa.

Apesar do *Google Custom Search* possibilitar o retorno do resultado nos formatos JSON e XML, optou-se pela adoção do primeiro por ser menos burocrático em sua estrutura, não sendo necessárias marcações no início e fim de cada sentença (como ocorre em XML), conforme observado por [Ghisi et. al 2010]. Isto permite uma redução no tamanho da resposta e, consequentemente, menor atraso na sua obtenção.

Finalmente, o retorno do resultado é enviado ao *gateway*, que o repassa para a parte da API presente na aplicação NCLua. Ao receber o resultado em JSON, a API faz uso da biblioteca JSON4Lua[6] para converter o resultado em uma tabela Lua, tornando-o assim consumível pela aplicação NCLua.

Com estes componentes de arquitetura, diversos cenários de integração TVDi – *Internet* se tornam possíveis. Os estudos de caso 1 e 2, apresentados na próxima seção, foram produzidos com o intuito de demonstrar o uso desta arquitetura em duas aplicações: 1) para o consumo de um *Web Service* por uma aplicação *t-government*

[6] http://json.luaforge.net

NCLua; e 2) para a integração de uma aplicação NCLua ao motor de buscas do Google e visualização de conteúdos em RSS, para enriquecimento de conteúdo televisivo.

6. Resultados Obtidos

Nesta seção são apresentadas as aplicações referentes aos estudos de caso realizados.

6.1. Estudo de Caso 1: SSP TV

Em [Arrais, Teixeira e Silva Filho 2012] foi produzido um protótipo de aplicação *t-government* NCLua, utilizando boa parte da arquitetura proposta: os módulos TCP, NCLua_HTTP, NCLua_SOAP e a API Leitor RSS. A aplicação teve como finalidade principal a integração de dois serviços governamentais, disponibilizados pela SSP-TO, para responder a algumas necessidades básicas da população, através de um veículo de comunicação presente em quase todas as residências (a TV): 1) integração ao sistema SIOP, o qual mantém informações de ocorrências policiais e de trânsito; 2) integração com a Delegacia Virtual, serviço para registros de boletins de ocorrência eletrônicos disponibilizado na *web*. A Figura 2 ilustra a funcionalidade de acompanhamento de Boletim de Ocorrência (BO) na aplicação em questão.

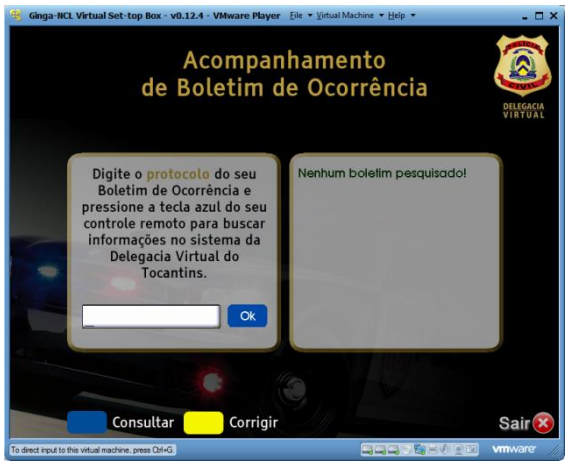

Figura 2. Estudo de Caso 1: Funcionalidade para Acompanhamento de BO

Esta imagem mostra parte da aplicação integrada ao serviço Delegacia Virtual. Com o módulo NCLua_SOAP, foi feito o consumo de um *Web Service* para permitir a consulta da situação de BOs eletrônicos, registrados via sistema *web* da Delegacia Virtual. Para a integração com o sistema SIOP, um leitor de RSS foi implementado na aplicação. O mesmo utiliza a API Leitor RSS para o consumo de informações, cadastradas no referido sistema, referentes a ocorrências de trânsito. Neste contexto, o RSS foi utilizado para prover um serviço de *e-gov* na TVDi. Apesar do consumo de dados na aplicação desenvolvida ter sido simulado, os serviços integrados são reais e estão disponíveis aos cidadãos tocantinenses.

6.2. Estudo de Caso 2: Uma Aplicação NCLua com Extensão de Conteúdo

Este estudo de caso, desenvolvido sobre a arquitetura proposta, resume-se a uma aplicação NCLua que utiliza a API *Google Search* para procurar *feeds* RSS na *Internet*, que podem ter relação com o conteúdo em exibição. Estes *feeds* são apresentados ao telespectador, o qual pode escolher qual deles deseja visualizar. Ao selecionar o *feed*, a aplicação NCLua, por meio da API Leitor RSS, busca o seu respectivo arquivo (XML)

na *Internet* e exibe o seu conteúdo ao usuário, provendo assim o enriquecimento do conteúdo televiso através das informações adicionais (RSS) advindas da *web*. A Figura 3 ilustra a aplicação NCLua desenvolvida.

Figura 3. Estudo de Caso 2: Extensão de Conteúdo em uma Aplicação NCLua

Quando o usuário de TV aciona a aplicação NCLua pelo controle remoto (Figura 3-A), é exibido um campo de entrada de dados (campo de texto) para pesquisas *web*, no qual o usuário pode informar palavras-chave relacionadas ao conteúdo em exibição ou qualquer assunto que deseja procurar. O ideal seria obter dados do EPG (palavras-chave que categorizam um determinado programa em exibição), para fazer a pesquisa automática sensível à programação corrente. Contudo, isso não foi possível em decorrência de dois motivos: 1) a aplicação reproduz um vídeo para simular a programação de uma emissora, ou seja, não estão sendo transmitidas as tabelas que contêm as informações do EPG; 2) o ambiente de simulação utilizado (Ginga-NCL Virtual STB), até o momento, não dispõe de recursos para acessar o EPG.

Com as palavras-chave informadas pelo usuário, a aplicação busca resultados no serviço *Google Custom Search* por meio da API *Google Search*, adicionando ao parâmetro de pesquisa uma restrição (filetype:rss) para retorno de resultados que apontem exclusivamente para arquivos com a extensão .rss. Neste momento, o usuário pode selecionar um dos resultados obtidos ou fazer uma nova pesquisa *web*. No caso de seleção de um dos resultados exibidos, a aplicação utiliza a API Leitor RSS para obter o arquivo XML na *Internet*, referente ao *feed* RSS selecionado. Com o *download* concluído, a aplicação exibe o conteúdo do RSS na tela da TV (Figura 3-B).

Com isso, o usuário pode buscar outras fontes de informações na *web*, que podem estar relacionadas e vão além do conteúdo televiso em exibição, e assim, obter mais detalhes de certo assunto, aprofundando o seu conhecimento sobre o mesmo.

7. Considerações Finais

Os dois estudos de caso apresentados mostram a utilização de todos os componentes (APIs) da arquitetura proposta em diversos cenários de integração TVDi – *Internet*: *download* de arquivos (neste trabalho, arquivos XML), integração com outros sistemas via HTTP e HTTPS e o consumo de *Web Services* por meio do protocolo SOAP.

Por outro lado, um ponto negativo da arquitetura proposta está na necessidade de utilização do *gateway* pela API *Google Search*, pois há concentração de requisições HTTP e, em se tratando de uma quantidade considerável de requisições simultâneas, a escalabilidade das aplicações que utilizam a API pode ser comprometida.

Trabalhos futuros podem ser realizados para aprimorar a arquitetura e as APIs desenvolvidas. Sobre a arquitetura, assim que o Ginga-NCL tiver suporte a requisições HTTPS, a camada intermediária (*gateway*) entre a API em NCLua e o serviço *Google*

Custom Search pode ser removida, já que o *gateway* foi utilizado apenas devido a necessidade de realizar requisições HTTPS. Acerca da API *Google Search*, como o serviço *Google Custom Search* oferece suporte para retorno nos formatos JSON e XML, pode-se implementar também a conversão do retorno em XML para uma tabela Lua. Sobre a API Leitor RSS, sugere-se estendê-la para suportar a leitura do formato ATOM, pois este formato também é baseado em XML e possui a mesma finalidade que o RSS, ou seja, publicação de conteúdo *web* que pode sofrer constantes atualizações.

Referências

ABNT. Associação Brasileira de Normas Técnicas (2008) 15606-1. "Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 1: Codificação de dados".

Arrais, E. Teixeira, T. S. e Silva Filho, M. C. da. (2012) "SSP Tv: Uma Aplicação para TV Digital Interativa baseada em NCLua" In: III EATI, 2012, Frederico Westphalen.

CETIC.Br. Centro de Estudos sobre as Tecnologias da Informação e da Comunicação. (2011). Disponível em: http://op.ceptro.br/cgi-bin/cetic/tic-domicilios-e-empresas-2011.pdf

Ferretti, S., Roccetti, M. e Palazzi, C. E. (2007) "Web Content Search and Adaptation for IDTV: One Step Forward in the Mediamorphosis Process toward Personal-TV". Advances in Multimedia, 2007.

Ghisi, B. C., Lopes, G. F. e Siqueira, F. (2010) "Integração de Aplicações para TV Digital Interativa com Redes Sociais". In: Webmedia, 2010.

Ginga-NCL. Site Oficial do Ginga-NCL. (2013) Disponível em: http://www.gingancl.org.br

Herrero, C., Cesar, P. e Vuorimaa, P. (2003) "Delivering MHP applications into a real DVB-T network: OtaDigi", In: Proceedings of the 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service. TELSIKS 2003. Vol. 1. IEEE.

IBGE. Instituto Brasileiro de Geografia e Estatística (2011) "Pesquisa Nacional por Amostras de Domicílio 2011 – PNAD". Disponível em: http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2011

Lima Jr., Eucássio G., Silva, J. M. L., Alves, A. O., Gonçalves, D. M., Viana, N. S. e Machado, M. C. (2010) "Convergência TVDi-Web: Extensão de API para Explorar a Interatividade no Acesso a Conteúdo Web Dinâmico". In: V CONNEPI, 2010, Maceió, Brasil.

Lua. Site Oficial da Linguagem Lua. (2013) Disponível em: http://www.lua.org

MC. Ministério das Comunicações. (2013) "Programa Nacional de Banda Larga (PNBL)". Disponível em: http://www.mc.gov.br/acoes-e-programas/programa-nacional-de-banda-larga-pnbl

NCL. Site Oficial da Linguagem NCL. (2013) Disponível em: http://www.ncl.org.br

Sant'Anna, F. (2013). Documentação NCLua. Disponível em: http://www.telemidia.puc-rio.br/~francisco/nclua

Silva Filho, M. C. da, Gondim, P. R. de L. (2011) "NCLua SOAP: Acesso a Web Services em aplicações de TVDi." In: WCGE, 2011, Florianópolis.