

**Universidade Federal da Paraíba**  
**Centro de Informática**  
**Programa de Pós-Graduação em Informática**

**UMA SOLUÇÃO PARA EXECUÇÃO DE APLICAÇÕES  
GINGA-NCL USANDO SEGUNDA TELA EM SISTEMAS  
*BROADBANDTV***

**Álan Livio Vasconcelos Guedes**

**Orientador: Guido Lemos de Souza Filho**

**João Pessoa**  
**Agosto/2012**

**Álan Livio Vasconcelos Guedes**

**UMA SOLUÇÃO PARA EXECUÇÃO DE APLICAÇÕES  
GINGA-NCL USANDO SEGUNDA TELA EM SISTEMAS  
*BROADBANDTV***

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

**Orientador: Prof. Dr. Guido Lemos de Souza Filho**

**João Pessoa**

**Agosto/2012**

# Resumo

No desenvolvimento de aplicações para TV Digital, o ambiente Ginga-NCL prove suporte a aplicações com uso de dispositivos de *segunda tela*. Entretanto, foi apenas com recentes esforços na implementação de referência do Ginga-NCL que esse recurso foi de fato viabilizado, utilizando uma estratégia de associação de perfis *UAProf* aos dispositivos necessários. Este trabalho propõe uma arquitetura de portal de aplicações *BroadbandTV* que utiliza o ambiente Ginga-NCL, sendo capaz de apresentar ao usuário uma lista de aplicações compatíveis ao seu contexto de execução multidispositivos. Como resultados, são apresentadas a descrição da arquitetura e a execução de testes com protótipos de software.

**Palavras chave:** TV Digital, NCL, aplicações multimídia distribuídas, *BroadbandTV* e *Application Store*.

# Abstract

On the development of applications for Digital the TV environment Ginga-NCL provides application support with use of second screen devices. However, it was only in fact made possible by recent efforts in the implementation of the Ginga-NCL reference using a strategy of association between *UAProf* profiles and the required devices. This work proposes architecture of Broadband TV Application Store that uses the Ginga-NCL, being able to present to user a list of compatible applications to their multidevice execution context. As results, are presented a description of the architecture and execution of tests with prototypes of software.

**Keywords:** Digital TV, NCL, distributed multimedia applications, *Broadband TV*, and *Application Store*

# Agradecimentos

Aos orientadores e amigos Guido Lemos e Carlos Batista, pela paciência e confiança depositada em mim.

A outros amigos igualmente orientadores Tatiana Tavares, Raoni Kulesza, Thiago Gouveia, Thiago Curvelo, Gedvan Dias, Erick Melo, Carlos Hacks, Gilberto Farias, Thiago Maritan, Rostand Costa e Hélio de Menezes.

Aos amigos de laboratório Fernando Brito, Ana Paula, José Ivan, Luís Felipe, Ruan Delgado, Lucenildo Lins, Thiago Cordinha, Júlio Cezar, Danilo Lima, Kellyane Alves, Cheylla entre outros pelo apoio direto e indireto para esta dissertação.

A minha família, dentre pais, irmãos, primos, tios e avós pelo amor e apoio incondicional.

A amigos pessoais tanto de infância, dos períodos de CEFET e UFPB, que de tantos não caberiam aqui, pela amizade e afeto.

Aos institutos financiadores de pesquisa CAPES CTIC e RNP por financiarem a pesquisa e outras pessoas que contribuíram para essa dissertação.

*In Memoriam* do amado tio Amaury dos Santos. Que me ensinou quanto o compromisso com trabalho torna o homem nobre e digno.

# Sumário

<b>Lista de Figuras .....</b>	<b>viii</b>
<b>Lista de Tabelas.....</b>	<b>x</b>
<b>1. Introdução .....</b>	<b>11</b>
<b>1.1. Motivação.....</b>	<b>11</b>
<b>1.2. Objetivos .....</b>	<b>13</b>
<b>1.3. Trabalhos no Contexto da Instituição .....</b>	<b>14</b>
<b>1.4. Organização da dissertação .....</b>	<b>15</b>
<b>2. Fundamentação Teórica .....</b>	<b>16</b>
<b>2.1. Descrições de características de dispositivos .....</b>	<b>16</b>
<b>2.2. TV Digital Interativa.....</b>	<b>18</b>
2.2.1. Arquitetura de sistemas de TV digital interativa .....	19
2.2.2. <i>Broadband TV</i> .....	21
2.2.3. Ginga-NCL.....	23
2.2.4. Aplicações Ginga-NCL com segunda tela .....	26
<b>3. Trabalhos Relacionados .....</b>	<b>29</b>
<b>4. Arquitetura de <i>software</i> da Solução .....</b>	<b>34</b>
<b>4.1. Cenário de uso e requisitos de alto nível .....</b>	<b>34</b>
<b>4.2. Subsistemas e requisitos de software .....</b>	<b>36</b>
<b>4.3. Visão de casos de uso .....</b>	<b>38</b>
<b>4.4. Visão de implantação .....</b>	<b>40</b>
<b>4.5. Visão de processos.....</b>	<b>42</b>
<b>4.6. Visão lógica.....</b>	<b>43</b>

<b>5. Implementação e testes .....</b>	<b>49</b>
<b>5.1. Visão de implementação .....</b>	<b>49</b>
<b>5.2. Casos de testes do sistema .....</b>	<b>53</b>
<b>5.3. Relatório de execução dos casos de testes.....</b>	<b>57</b>
<b>6. Considerações Finais .....</b>	<b>60</b>
<b>6.1. Trabalho desenvolvido.....</b>	<b>60</b>
<b>6.2. Contribuições.....</b>	<b>61</b>
<b>6.3. Trabalhos futuros .....</b>	<b>62</b>

# Lista de Figuras

Figura 1: Aplicação de segunda tela.....	12
Figura 2: Interface da Samsung <i>Smart TV</i> .....	13
Figura 3: Integração de cenários com heterogeneidade multidispositivos.....	13
Figura 4: Descrição de componentes de um dispositivo móvel com <i>UAProf</i> .....	17
Figura 5: Visão de alto nível de um sistema de TV Digital.....	19
Figura 6: Arquitetura em camadas no receptor de TV Digital.....	19
Figura 7: Camada de <i>Middleware</i> em um Receptor de TV Digital.....	21
Figura 8: Arquitetura de sistemas IPTV.....	22
Figura 9: A suíte de especificações MARF.....	25
Figura 10: Integrando exibidores de mídia de no ambiente Ginga-NCL.....	25
Figura 11: Telas da aplicação de controle de segunda tela do <i>Yahoo Connected TV</i> .....	30
Figura 12: Tela da aplicação com uso de segunda tela do GoogleTV.....	31
Figura 13: Sinalização do <i>Sticker Center</i> a dispositivos de segunda tela.....	32
Figura 14: Descrição ontológica da Physical Device.....	32
Figura 15: Visão geral do sistema.....	34
Figura 16: Instância de cenário de uso com parâmetros suas aplicações e contextos de execução multidispositivos.....	35
Figura 17: Diagrama de casos de uso do subsistema <i>GingaStore</i> .....	39

<b>Figura 18: Diagrama de casos de uso do subsistema <i>GingaSpace</i> .....</b>	<b>40</b>
<b>Figura 19: Diagrama de implantação do sistema .....</b>	<b>41</b>
<b>Figura 20: Diagrama de colaboração entre atores do sistema.....</b>	<b>43</b>
<b>Figura 21: Um descrição de contexto de execução <i>Ginga-NCL</i> de perfil EDTV e suporte a mídias <i>HTML5</i>, e outro de perfil EDTV conectividade <i>UPnP</i> .....</b>	<b>45</b>
<b>Figura 22: Visão geral da descrição de requisitos de aplicação no <i>GingaSpace</i> .....</b>	<b>45</b>
<b>Figura 23: Diagrama de estados do <i>GingaSpace</i>.....</b>	<b>47</b>
<b>Figura 24: Semântica da conectividade de dispositivos no <i>GingaStore</i> .....</b>	<b>48</b>
<b>Figura 25: Diagrama de pacotes do <i>GingaStore</i>.....</b>	<b>50</b>
<b>Figura 26: Diagrama de pacote do <i>GingaSpace</i>.....</b>	<b>51</b>
<b>Figura 27: Protótipo do Portal <i>Web GingaSpace</i> .....</b>	<b>52</b>
<b>Figura 28 - Execução do protótipo do <i>GingaStore</i>.....</b>	<b>53</b>
<b>Figura 29: Configuração de equipamentos para execução dos testes .....</b>	<b>59</b>
<b>Figura 30: <i>GingaAppStore PlayList</i> e configuração de <i>EITV Playout</i>.....</b>	<b>62</b>

# Lista de Tabelas

<b>Tabela 1: Ambientes de <i>Broadband TV</i>.</b> .....	<b>23</b>
<b>Tabela 2: Definição da classe <i>DeviceGroup</i>.</b> .....	<b>27</b>
<b>Tabela 3: Atributos adicionados às classes <i>UAProf</i>.</b> .....	<b>27</b>
<b>Tabela 4: Requisitos relacionados ao <i>GingaStore</i>.</b> .....	<b>37</b>
<b>Tabela 5: Requisitos relacionados repositório <i>GingaSpace</i>.</b> .....	<b>38</b>
<b>Tabela 6: Casos de testes de cadastro de aplicações no <i>GingaSpace</i>.</b> .....	<b>53</b>
<b>Tabela 7: Casos de testes de <i>GingaStore</i> recuperando de lista de aplicações compatíveis no <i>GingaSpace</i>.</b> .....	<b>55</b>
<b>Tabela 8: Casos de testes de da aplicação <i>GingaStore</i>.</b> .....	<b>56</b>
<b>Tabela 9: Configurações de equipamentos utilizado na execução dos testes sobre o sistema</b> .....	<b>57</b>
<b>Tabela 10: Comparação com trabalhos relacionados considerando o <i>GingaSpace</i>.</b> .....	<b>61</b>

# 1. Introdução

Segundo Fernandes et al. (2004) a transição da TV analógica para a TV digital permitiu novas características as transmissões de TV. É possível destacar como principais características a qualidade de som e imagem, mobilidade de recepção, multiprogramação de canais de transmissão *broadcast* e a interatividade. A interatividade na TV Digital configura-se como a área de pesquisa desta dissertação.

O recuso de interatividade possibilita a execução nos receptores de TV de programas de computador em conjunto ao conteúdo audiovisual tradicional. Segundo Becker (2006), o surgimento da interatividade quebra o paradigma de programas de vídeo lineares, permitindo uma maior participação do usuário na escolha da formatação do conteúdo. A interatividade permite o usuário a abandonar a passividade, característica inerente na TV analógica, e passe a interagir de diversas formas com o conteúdo transmitido.

## 1.1. Motivação

Este trabalho tem o intuito de incrementar o ecossistema de interatividade na TV Digital, abordando dois recentes cenários de uso: o de aplicações de *second screen* (CRUICKSHANK, 2007), e o de aplicações para receptores de *BroadbandTV* (as TV Conectadas) (GELLNSKI, 2012).

O cenário de aplicações de *segunda tela* possibilita criar uma apresentação multimídia distribuída entre a TV e os dispositivos em uma rede doméstica, como celulares, *tablets*, entre outros. A Figura 1 ilustra a execução de uma aplicação de uso de segunda tela. Nela, é exibida uma execução de um jogo de quebra cabeças no dispositivo de segunda tela sincronizado com o conteúdo audiovisual exibido pelo receptor de TV.

Diferentes especificações de sistemas de TV Digital terrestre fornecem interfaces de programação para o desenvolvimento desse tipo de cenário. Entretanto, essas especificações não definem como deve ser o comportamento do receptor para realizar a descoberta e a comunicação

com os dispositivos de *segunda tela*. Logo, dada à **heterogeneidade dos dispositivos de segunda tela**, as presentes especificações delegam aos fabricantes dos receptores que implementem seus próprios mecanismos de comunicação.

**Figura 1: Aplicação de segunda tela.**



Fonte: (ITU, 2012).

Avanços recentes propostos para o *middleware* Ginga (ABNT, 2008b) permitiram uma melhor descrição da lógica de cooperação entre receptores de TV e dispositivos de segunda tela. O trabalho de Batista et al. (2010) utiliza uma estratégia não restritiva para que o autor de uma aplicação interativa para TV Digital descreva quais características dos dispositivos de *segunda tela* são necessárias, facilitando a interoperabilidade com diferentes tipos de dispositivos.

O cenário de receptores *BroadbandTV* realiza uma integração entre o receptor de TV e serviços da *Internet*. Nesse cenário tanto o conteúdo audiovisual quanto aplicações são recuperadas por meio de conectividade com a *Internet*. Ele se utiliza da popularização da banda larga e serviços de conteúdo sobre demanda da *Web* como *Youtube*<sup>1</sup> e *Netflix*<sup>2</sup>. A Figura 2 apresenta a interface de seleção de serviços de Internet da *Samsung Smart TV*<sup>3</sup>.

Para viabilizarem a integração com a Internet, os receptores *Broadband TV* utilizam tecnologias da *Web* como *HTML5*<sup>4</sup>, *Adobe Flash*<sup>5</sup>, *JavaScript*<sup>6</sup>, entre outras. Cada fabricante de receptores de *Broadband TV* possui interfaces de programação proprietárias, fazendo com que as aplicações sejam dependentes de diferentes plataformas.

---

<sup>1</sup> <http://www.youtube.com>. Acessado em: agosto de 2012.

<sup>2</sup> <http://www.netflix.com>. Acessado em: agosto de 2012.

<sup>3</sup> <http://www.samsung.com.br/smarttv>. Acessado em: agosto de 2012.

<sup>4</sup> <http://www.w3.org/TR/2011/WD-html5-20110525>. Acessado em: agosto de 2012.

<sup>5</sup> <http://www.adobe.com/br/products/flash.html>. Acessado em: agosto de 2012.

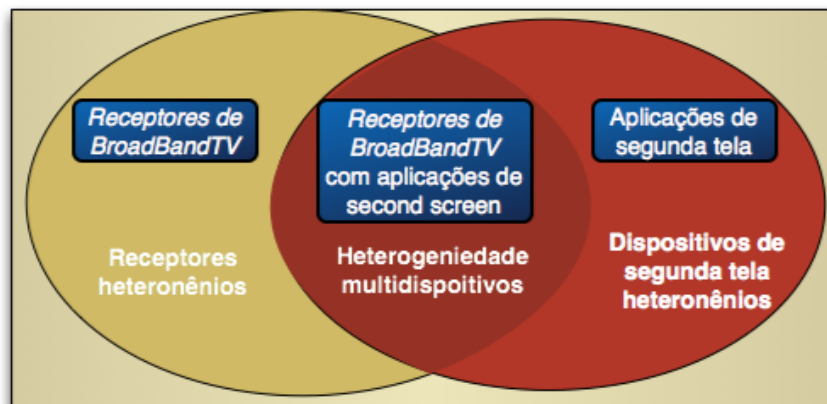
<sup>6</sup> <http://www.w3.org/standards/techs/js>. Acessado em: agosto de 2012.

**Figura 2: Interface da Samsung Smart TV.**



Esta dissertação aborda um cenário que integra esses dois cenários de uso da TV Digital Interativa. A integração é caracterizada por uma heterogeneidade tanto dos receptores *Broadband TV* quanto dos dispositivos de segunda tela. A Figura 3 oferece uma abstração para a integração dos cenários de uso citados.

**Figura 3: Integração de cenários com heterogeneidade multidispositivos.**



## 1.2. Objetivos

Este trabalho propõe uma solução para execução de aplicações de segunda tela em sistemas de *BroadBandTV* para tratar a heterogeneidade multidispositivos citada na sessão anterior. Essa solução se baseia no ecossistema de Ginga-NCL e em uma descrição de dispositivos baseada em BATISTA (2010), que permite aos autores de aplicações definirem os seus requisitos frente à

heterogeneidade citada, e aos usuários encontrar quais aplicações podem ser executadas em seu dispositivo.

Os seguintes objetivos específicos foram definidos para a obtenção do objetivo geral:

- Apresentação de uma fundamentação teórica em descrições de requisitos de dispositivos;
- Apresentação de uma fundamentação teórica em *Broadband TV* e aplicações de segunda tela;
- Proposição de uma arquitetura de repositório de aplicações para receptores *Broadband TV*;
- Implementação de um protótipo da arquitetura proposta;
- Avaliação da execução de casos de teste com intuito de validar a arquitetura proposta;
- e.
- Evidenciar a contribuição aos usuários finais e ao ecossistema *Broadband TV*.

### 1.3. Trabalhos no Contexto da Instituição

A participação na execução de dois projetos de pesquisa permitiu criar uma fundamentação técnica e teórica para o desenvolvimento dessa dissertação. Ambos executados dentro do LAViD – Laboratório de Aplicações de Vídeo Digital<sup>1</sup> durante período da pós-graduação.

O primeiro projeto, o *GingaCDN - GINGA Code Development Network*<sup>2</sup>, foi realizado no período de 2009 e 2010, e teve o intuito de criar uma rede de desenvolvedores de componentes e aplicações para o *middleware* brasileiro de TV Digital, permitindo o desenvolvimento aberto, distribuído e colaborativo da implementação do *middleware* OpenGinga<sup>3</sup>. Oferecendo um código sobre a licença *GNU GPL - General Public License*<sup>4</sup> de um sistema compatível com a especificação de receptores fixos compatíveis com o SBTVD<sup>5</sup>, ou seja, com suporte a aplicações Ginga-J e Ginga-NCL.

---

<sup>1</sup> <http://www.lavid.ufpb.br>. Acessado em: agosto de 2012.

<sup>2</sup> <http://openginga.net>. Acessado em: agosto de 2012.

<sup>3</sup> <http://openginga.net/projetc/openginga>. Acessado em: agosto de 2012.

<sup>4</sup> <http://www.gnu.org/copyleft/gpl.html>. Acessado em: agosto de 2012.

<sup>5</sup> <http://www.forumsbtvd.org.br>. Acessado em: agosto de 2012

O segundo projeto, o *GingaAppStore* (DE SOUZA FILHO, 2009), foi realizado no período de 2011 a 2012 e tem o intuito de potencializar a capacidade de interatividade do SBTVD ao desenvolver ferramentas que auxiliem no processo de desenvolvimento, disponibilização e divulgação de aplicações. Apresentando uma implementação de um repositório público de aplicações interativas para receptores do SBTVD. Utilizando como estudo de caso uma aplicação de T-Health (Becker, 2006) para controle de vacinação, chamada *ZeGotinhaTV*, que utiliza uma arquitetura para canal de interação a partir da rede pública do NavegaPará<sup>1</sup>.

Enquanto a execução do projeto GingaCDN ofereceu fundamentação técnica e teórica acerca de requisitos sobre de *middleware* de TV Digital, o projeto *GingaAppStore* ofereceu no cenário de *Broadband TV*.

## 1.4. Organização da dissertação

Esta dissertação está organizada em seis capítulos. Após a introdução, os seguintes capítulos são apresentados:

- **No Capítulo 2** é apresentada a fundamentação teórica, com conceitos que permitem melhor entendimento da dissertação;
- **No Capítulo 3** são apresentados trabalhos relevantes no do estado da arte abordado por esta dissertação;
- **No Capítulo 4** é apresentado um cenário de uso para a solução proposta. Juntamente com a descrição da solução através da especificação da sua arquitetura de *software*;
- **No Capítulo 5** são apresentados detalhes de implementação de um protótipo da arquitetura, mais os casos de testes e quais avaliações foram conduzidas para validação da solução proposta.
- **No Capítulo 6** são apresentadas as considerações acerca do que foi realizado no trabalho, evidenciando as contribuições e também possíveis perspectivas de evolução para o trabalho realizado durante a elaboração desta dissertação.

---

<sup>1</sup> <http://www.navegapara.pa.gov.br>. Acessado em: agosto de 2012.

## 2. Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica com os principais conceitos abordados nesta dissertação. Inicialmente são abordados conceitos relacionados aos mecanismos de descrição das características de dispositivos computacionais. Em seguida é apresentada uma introdução da TV Digital Interativas, e de seus cenários de receptores *Broadband TV* e uso de aplicações de *segunda tela*, com um ênfase no ambiente de execução Ginga-NCL.

### 2.1. Descrições de características de dispositivos

Mecanismos para descrição de dispositivos são necessários para a criação de sistemas distribuídos sistemas sensíveis a contexto de execução. Segundo Schmidt (1994) um contexto pode ser definido como qualquer informação que pode ser utilizada para representar a interação entre o usuário e o sistema. Esses sistemas devem ser capazes de representar de forma padronizada os serviços e características dos dispositivos, de modo a permitir a recuperação e processamento dessas informações por entidades de *software*.

Um importante esforço na construção de mecanismos de representação dessas informações é realizado pelo consórcio internacional W3C<sup>1</sup>. Ele propõe o uso de documentos XML<sup>2</sup> e RDF<sup>3</sup> para construir ferramentas de representação das informações na Web, constituindo uma *Web Semântica* (BERNERS-LEE, 2001).

A descrição *UAProf* - User Agent Profile (*OPEN MOBILE ALLIANCE*, 2001) é uma implementação concreta do framework *W3C CC/PP* (*Composite Capabilities/Preference Profiles*<sup>4</sup>), que utiliza documentos RDF para descrever capacidades e características de dispositivos móveis. Ele descreve características de *hardware* (como tela e processador),

---

<sup>1</sup> <http://www.w3.org>. Acessado em: agosto de 2012.

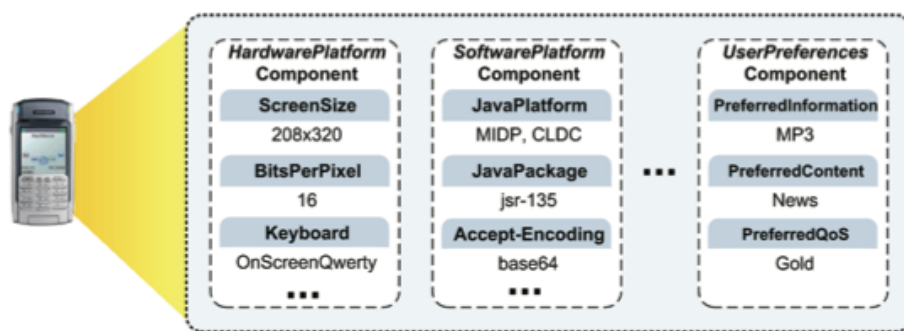
<sup>2</sup> <http://www.w3.org/XML>. Acessado em: agosto de 2012.

<sup>3</sup> <http://www.w3.org/RDF>. Acessado em: agosto de 2012.

<sup>4</sup> <http://www.w3.org/Mobile/CCPP>. Acessado em: agosto de 2012.

*software* (como versão de *Java*<sup>1</sup> suportada, exibidores de mídia etc.), características de rede (como capacidades de *GSM/GPRS* e *Bluetooth*<sup>2</sup>), características de navegador nativo (como versões suportadas de *HTML*, *Java script*) etc. A Figura 4 exemplifica uma descrição *UAProf* para um dispositivo móvel.

**Figura 4: Descrição de componentes de um dispositivo móvel com *UAProf*.**



Fonte: (FRKOVIC et al., 2008).

A especificação *UAProf* foi originalmente concebida para que as informações do perfil do dispositivo fossem enviadas junto ao cabeçalho de suas requisições HTTP, de forma que o conteúdo a ser recebido pudesse ser adaptado pelo servidor HTTP respondendo à requisição. Ressalta-se o uso de repositórios alternativos aos de fabricantes de mobile, como é o caso da comunidade *WURFL - Wireless Universal Resource FiLe*<sup>3</sup>.

O *OSGi - Open Services Gateway Initiative* (*OSGi*, 2012) define uma plataforma de serviços baseada no ambiente de programação *Java*, implementando um dinâmico modelo de componentes de *software* distribuídos. Os componentes, chamados *bundle*, são remotamente instalados, iniciados, parados, atualizados e desinstalados em dispositivos. O gerenciamento de ciclo de vida dos *bundles* (iniciar, parar, instalar) é realizado através de uma API *Java*. Além de registrar seus serviços, cada *bundle* tem a capacidade de detectar e invocar serviços de outros *bundle*.

O *UPnP* (*UPNP FÓRUM*, 2012) propõe um framework para construir aplicações em rede, que permite a integração dos serviços fornecidos por dispositivos em uma rede doméstica. Dispositivos *UPnP* fornecem indicações de seus serviços através de *URL* dentro de descrições

<sup>1</sup> <http://www.jcp.org>. Acessado em: agosto de 2012.

<sup>2</sup> <http://www.bluetooth.org>. Acessado em: agosto de 2012.

<sup>3</sup> <http://wurfl.sourceforge.net>. Acessado em: agosto de 2012.

*XML*; ao recuperar uma descrição de dispositivo, um ponto de controle pode enviar ações de serviço para um dispositivo utilizando mensagens no formato *SOAP - Simple Object Access Protocol*<sup>1</sup>. Uma descrição de serviço *UPnP* define ações suportadas e seus argumentos, variáveis de estado e dados arbitrários. O fórum *UPnP* desenvolveu especificações para algumas classes de dispositivos, os chamados *DPC - Device Control Protocol*. Cada *DCP* define uma interface comum para uma classe de dispositivos *UPnP* com serviços incorporados em termos de ações obrigatórias e opcionais e variáveis de estado. Pontos de controle que têm conhecimento prévio do *DCP* que um dispositivo está usando, pode, portanto, facilmente fazer uso de sua interface.

## 2.2. TV Digital Interativa

No final da década de 80, pesquisadores japoneses realizaram estudos para criar um sistema de distribuição de conteúdo audiovisual de alta resolução (*HDTV - High Definition Television*) (NHK-STRL 2012). Esses estudos foram o ponto de partida para a especificação de sistemas de TV digital terrestre nos Estados Unidos<sup>2</sup>, Europa<sup>3</sup> e no Japão<sup>4</sup>.

Segundo Fernandes et al. (2004), a transição para TV Digital gera três mudanças essenciais. A primeira mudança é a produção de conteúdo digital através do uso de novos equipamentos de captura e edição digital. A segunda é o novo modelo de transmissão e recepção utilizado, através da adoção de padrões de codificação e difusão de sinais digitais, que permitem o envio de um volume maior informações e, assim, a transmissão da programação audiovisual com melhor qualidade (resolução) e de aplicativos interativos. A terceira mudança acontece nos modelos de negócios que financiam a cadeia de produção da TV Digital, com a necessidade de se fomentar o interesse do consumidor pelo novo formato e novos mecanismos que aumentam a rentabilidade do negócio para os produtores e difusores de conteúdo de TV Digital.

As subseções seguintes descrevem a arquitetura de um sistema de TV digital Interativa, características dos receptores de *Broadband TV*, informações sobre o *middleware* Ginga e sobre aplicações de TV Digital interativa que utilizam o conceito de segunda tela.

---

<sup>1</sup> <http://www.w3.org/TR/soap>. Acessado em: agosto de 2012.

<sup>2</sup> <http://www.atsc.org>. Acessado em: agosto de 2012.

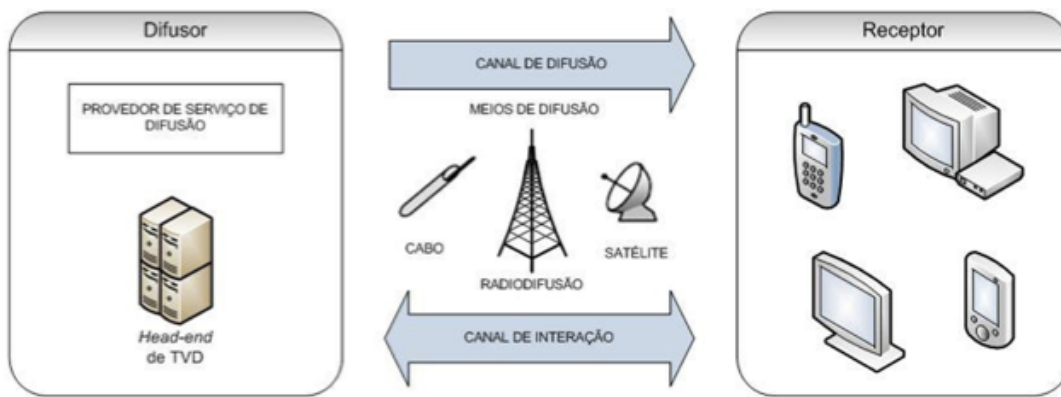
<sup>3</sup> <http://www.dvb.org>. Acessado em: agosto de 2012.

<sup>4</sup> <http://www.dibeg.org> Acessado em: agosto de 2012.

### 2.2.1. Arquitetura de sistemas de TV digital interativa

Segundo Becker (2006) a TV digital Interativa pode ser modelada de forma genérica com a composição de quatro entidades ilustradas na Figura 5: o **difusor**, responsável por transmitir o conteúdo áudio visual e dados para os receptores de TV; o **canal de difusão**, que é o canal de comunicação por difusão para os receptores, usado para distribuição do conteúdo; o **receptor**, que é o dispositivo que consome o conteúdo de TV; e o **canal de interação**, que é um canal de comunicação que permite troca de informações entre o receptor e o difusor.

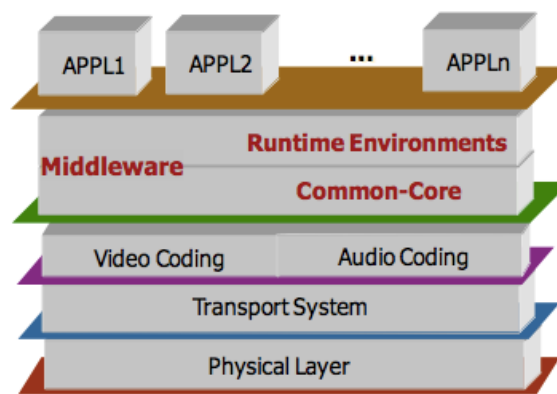
**Figura 5: Visão de alto nível de um sistema de TV Digital.**



Fonte: (BATISTA et al., 2008).

Assim como no projeto de infraestrutura de redes de computadores, o projeto de um sistema de TV digital utiliza a estratégia de serviços em camadas para reduzir sua complexidade. A Figura 6 ilustra o uso dessa estratégia no receptor, permitindo que cada camada ofereça serviços à camada imediatamente superior, e utilize os serviços da camada inferior.

**Figura 6: Arquitetura em camadas no receptor de TV Digital.**



Fonte: (SOARES, 2010).

A **Camada de física** é responsável por estabelecer a comunicação física entre o difusor e o receptor e deve estar presente em ambos. Exemplos dessa comunicação são a radiodifusão terrestre, transmissão por satélite ou cabo. A **Camada de transporte** é responsável pelo transporte do conteúdo audiovisual e dados entre o difusor e receptor, e deve estar presente em ambos. Dentro dos sistemas de TV Digital terrestre a tecnologia mais difundida nessa camada é padrão *MPEG-2 System* do grupo MPEG<sup>1</sup>, enquanto os ambientes IPTV utilizam protocolo de transporte *IP*<sup>2</sup>.

Segundo Soares (2010), os sistemas de TV Digital têm sido relatados na literatura com diferentes denominações: TV Digital terrestre, TV Digital a cabo, TV Digital satélite, IPTV, *Web TV*, *Broadband TV*, TV P2P, entre outros. Entretanto, esses termos são empregados dependendo basicamente da plataforma de transporte utilizada. Essas características de infraestrutura de transporte tornam os sistemas diferentes com relação aos serviços que oferecem. Por exemplo, serviços de VOD - vídeo sobre demanda, que são à base dos sistemas de IPTV, são dificilmente fornecidos em TV digital terrestre.

A **Camada de codificação** é responsável por realizar os processos de codificação do conteúdo audiovisual no difusor e decodificação nos receptores. Sendo esta camada usualmente implementada em *hardware*. Enquanto a **Camada de Middleware** oferece um serviço padronizado para a execução das aplicações interativas no receptor. Ela considera o fato do mercado TV Digital ser caracterizado por diferentes fabricantes de receptores. Ilustrado na Figura 7, o *middleware* viabiliza a interoperabilidade das aplicações a serem executadas em diferentes receptores, com configurações distintas de hardware e sistema operacional.

A **Camada de aplicação** é formada pelas aplicações TV que utilizam os recursos oferecidos pelo *middleware*. Segundo Becker (2006) a concepção de aplicações de TV possui peculiaridades que as diferem do desenvolvimento de aplicações em computadores. Dentre elas destacam-se resolução de tela, a distância entre o usuário e a tela, componentes gráficos, apresentação coletiva, hardware limitado, entrada de dados utilizando controle remoto, entre outros. As aplicações de TV Digital também apresentam peculiaridades que as diferem dos

---

<sup>1</sup> <http://www.mpeg.org>. Acessado em: agosto de 2012.

<sup>2</sup> <http://www.ietf.org/rfc/rfc791.txt>. Acessado em: agosto de 2012.

programas de televisão tradicionais, como a não narrativa linear. Por estes motivos não existe um consenso na literatura de definição precisa para aplicação interativa.

**Figura 7: Camada de *Middleware* em um Receptor de TV Digital.**



Fonte: (SILVA, 2010).

Alguns autores como Gawlinski (2003) associam as aplicações de TV interativa ao conceito de diálogo entre produtor de conteúdo e espectador. Enquanto autores como (CHORIANOPOULOS ,2004) e LU (2005) descrevem as aplicações como serviços interativos que enriquecem o conteúdo, tornando a experiência do telespectador mais participativa.

As aplicações de TV digital interativa criam um novo tipo de mídia a ser utilizado não apenas pelos produtores de conteúdo, desenvolvedores de jogos, ações do governos, entre outros. Elas podem ser aplicadas em diferentes contextos: Enquetes ou *TV Voting* (KULESZA ,2011); Comercio eletrônico ou *T-Commerce* (GHISI 2010a); educação à distância ou *T-Learning* (Oliveira et al., 2010), serviços de saúde ou *T-Health* (Oliveira et al., 2010b); serviços governamentais ou *T-Gov* (DA SILVA, et al., 2010), serviços financeiros ou *T-Banking* (VANDEKAR 2007); jogos ou *T-Game* (SEGUNDO et al., 2010), integração com redes sociais (GHISI, 2010b); entre outros.

### **2.2.2. *Broadband TV***

Para construir o conceito de *BroadBrand TV* é necessario primeiramente apresentar os sistemas de IPTV. O ITU (2012) define IPTV como serviços multimídia para televisão com video, áudio, texto gráficos e dados fornecidos sobre redes baseados em protocolo IP<sup>1</sup> capazes de fornecer o nível de qualidade exigido em serviços de TV tradicional. Esses serviços englobam

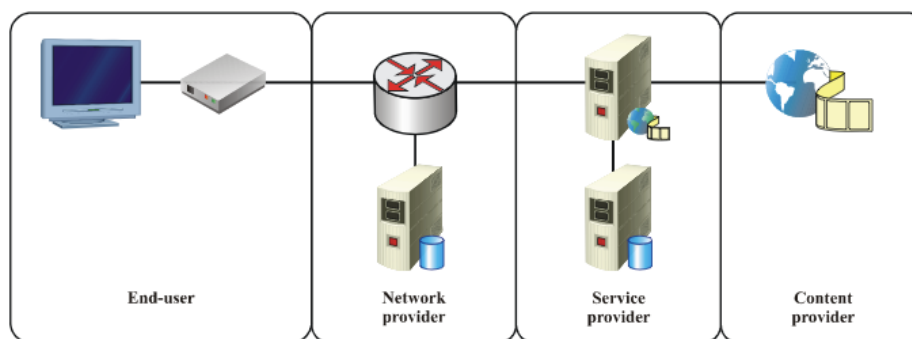
---

<sup>1</sup> <http://www.ietf.org/rfc/rfc791.txt>. Acessado em: agosto de 2012.

TV tradicional e serviços que envolvem uma combinação de comunicação e conteúdo sobre demanda, como VoIP – voz sobre *IP* e VoD - vídeo sob demanda. A Figura 8 ilustra a arquitetura de sistemas IPTV, em que o usuário final consome o conteúdo dos provedores que serviços através de provedores de rede IP com garantia de qualidade de serviço.

Segundo Soares (2010) a partir desses sistemas de IPTV foi possível viabilizar os receptores *BroadBrand TV*, ou seja, dispositivos com acesso de banda larga a Internet. Nesses receptores os provedores de conteúdo dividem a atenção do público entre tarefas assistir vídeos, navegar e mandar mensagens. Para Kulesza et al. (2011) essas tarefas são simplificadas pelo uso de aplicações chamadas *widjets*, que são executadas simultaneamente com os programas de TV.

**Figura 8: Arquitetura de sistemas IPTV.**



Fonte: (ITU 2009c).

Para a disponibilização dessas aplicações, os receptores *Broadband TV* reforçam a conceito de “portal de aplicações” (QUICO, 2003) dos primeiros serviços de TV Digital a cabo, através do uso de lojas de aplicações, ou *Application Store* (FRID, 2011).

Segundo Sawant (2010) o uso de *Application Stores* foi bem difundido no ecossistema dos dispositivos móveis ao agregar valor aos dispositivos criando um canal de varejo para distribuição de aplicações aos consumidores finais. Dentre estas lojas podemos citar a *iPhone AppStore*<sup>1</sup> da *Apple*, a *OVI*<sup>2</sup> da *Nokia* e o *Google Play*<sup>3</sup> da *Google*. De modo semelhante, as *Application Stores* nos receptores *Broadband TV* criam oportunidades de negócio para os fabricantes de receptores, que antes só eram exploradas pelos provedores de conteúdo. Nesse modelo, as aplicações usualmente utilizam recursos específicos de cada receptor, criando uma

<sup>1</sup> <http://apple.com/iphone/apps-for-iphone>. Acessado em: agosto de 2012

<sup>2</sup> <http://store.ovi.com>. Acessado em: agosto de 2012

<sup>3</sup> <http://play.google.com/store>. Acessado em: agosto de 2012

heterogeneidade de recursos e serviços que tem o intuito de agregar valor aos receptores *Broadband TV*, fomentando a concorrência entre eles.

Exemplos de *Application Store* de *Broadband TV* são a *YahooTV*<sup>1</sup> da Yahoo, o *Sticker Center*<sup>2</sup> da *TOTVS* (TOTVS, 2010), *SmartTV*<sup>3</sup> da *Samsung*, *NetTV* da *Philips*<sup>4</sup>, *Internet TV* da *Sony*<sup>5</sup>, *VieraCast*<sup>6</sup> da *Panasonic*. A Tabela 1 apresenta um comparativo dessas *Application Store* e de seus formatos de aplicações.

**Tabela 1: Ambientes de *Broadband TV*.**

<i>Empresa</i>	<i>Nome da plataforma</i>	<i>Vinculado à fabricante</i>	<i>Serviço de Application Store</i>	<i>Formato de aplicações</i>
<i>LG</i>	<i>NetCast</i>	Sim	Aplicações selecionadas	Sem <i>SDK</i> disponível
<i>Panasoni</i>	<i>VieraCast</i>	Sim	Aplicações selecionadas	Sem <i>SDK</i> disponível
<i>Philips</i>	<i>NetTV</i>	Sim	Aplicações selecionadas	CE-HTML
<i>Sony</i>	<i>InternetTV</i>	Sim	Aplicações selecionadas	Sem <i>SDK</i> disponível
<i>Sharp</i>	<i>AquousNet</i>	Sim	Aplicações selecionadas	Sem <i>SDK</i> disponível
<i>TOTVS</i>	<i>Sticker</i>	Sim	Sim	Lua
<i>Google</i>	<i>Google TV</i>	Não	Sim	<i>Android</i>
<i>Yahoo</i>	<i>Yahoo</i>	Não	Sim	<i>JavaScript</i>

Fonte: Adaptado de (FRID, 2011).

### 2.2.3. Gíngua-NCL

O Brasil adotou o SBTVD - Sistema Brasileiro de Televisão Digital em 2007. A partir da sanção de um decreto do Presidente da República que dispõe sobre a implantação do SBTVD, estabeleceram-se as diretrizes para a transição do serviço de radiodifusão analógica para o modelo digital, além da criação do seu órgão gestor, o Fórum SBTVD.

Por ter sido especificado recentemente, o SBTVD acumulou as experiências e decisões dos outros padrões. Ele segue a linha do ISDB-T japonês, privilegiando a alta definição e a recepção móvel, sendo também conhecido como ISDB-Tb. Sua principal mudança se dá na camada de

<sup>1</sup> <http://tv.yahoo.com>. Acessado em: agosto de 2012

<sup>2</sup> <http://stickercenter.com.br>. Acessado em: agosto de 2012

<sup>3</sup> <http://www.samsung.com.br/smarttv>. Acessado em: agosto de 2012

<sup>4</sup> <http://www.nettv.philips.com>. Acessado em: agosto de 2012

<sup>5</sup> <http://www.sony.co.uk/hub/internet-tv>. Acessado em: agosto de 2012

<sup>6</sup> <http://www.vieraexperience.com/html>. Acessado em: agosto de 2012

*middleware* ao utiliza o Ginga<sup>1</sup>, considerado a principal inovação fruto dos esforços dos laboratórios LAVID<sup>2</sup> da UFPB e Telemídia<sup>3</sup> PUC-Rio.

O Ginga é um *middleware* para receptores de TV Digital Terrestre e de IPTV. Ele possui como base o subsistema lógico do Ginga-NCL (SOARES, 2007) e pode possuir outros subsistemas como extensões. Por exemplo, o ITU defini extensões para a implementação específicos para IPTV e *Broadband TV* como VOD, entre outros.

O Ginga-NCL é constituído pelo ambiente de apresentação e aplicações declarativas escritas na linguagem NCL. Segundo Soares (2007) o NCL realiza uma rigorosa separação entre conteúdo e estrutura da apresentação multimídia, atuando como uma linguagem cola que conecta e sincroniza mídias, mas não restringe os tipos de mídia que utiliza.

O ISDB-Tb especifica o Ginga seguindo a orientação da recomendação J.200 (ITU, 2001) do ITU - *International Telecommunication Union*<sup>4</sup> e o utiliza o Ginga-J (DE SOUZA FILHO, 2007) como subsistema de extensão. Segundo Kulesza et al. (2010) o Ginga-J é um ambiente de execução de aplicações procedurais *Java* constituído por: uma *JVM – Java Virtual Machine* com perfis *CDC - Connected Device Configuration*<sup>5</sup> e *FP - Foundation Profile*<sup>6</sup>; a *JavaTV* (SUN MICROSYSTEMS, 2008), presente em outros ambientes procedurais de TV; e *JavaDTV* (SUN MICROSYSTEMS, 2008b), que consiste num conjunto de *API* que substituí funcionalmente o *GEM*<sup>7</sup>.

Segundo Moreno (2010), em sistemas de IPTV o ITU especifica o Ginga através da série de recomendações H.760 (ITU, 2009a). A série define o framework de formatos de aplicações para IPTV chamado MAFR - *Multimedia Application Framework*, ilustrado na Figura 9. Ela define o Ginga- NCL como uma ferramenta de cola e interoperabilidade entre os formatos de aplicações do MARF, devido a sua característica de não restrição às mídias que utiliza.

---

<sup>1</sup> <http://ginga.org.br>. Acessado em: agosto de 2012

<sup>2</sup> <http://www.lavid.ufpb.br>. Acessado em: agosto de 2012

<sup>3</sup> <http://www.telemidia.puc-rio.br>. Acessado em: agosto de 2012

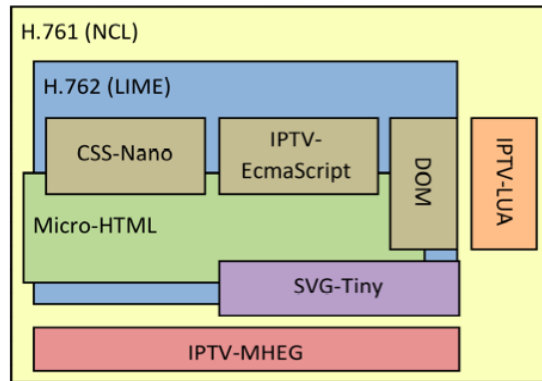
<sup>4</sup> <http://www.itu.int>. Acessado em: agosto de 2012

<sup>5</sup> <http://jcp.org/en/jsr/detail?id=218>. Acessado em: agosto de 2012

<sup>6</sup> <http://jcp.org/en/jsr/detail?id=219>. Acessado em: agosto de 2012

<sup>7</sup> <http://www.etsi.org>. Acessado em: agosto de 2012

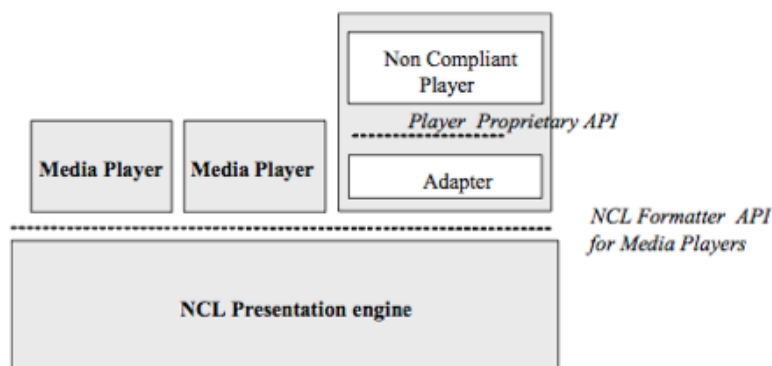
**Figura 9: A suíte de especificações MARF.**



Fonte: (MORENO, 2010).

Dois documentos complementam essa série de recomendações: a H.761 (ITU, 2009b), que atua de modo semelhante a (ABNT, 2008b) definindo o perfil EDTV e o formato de aplicações NCL para o ambiente Ginga-NCL em receptores IPTV. Ressalta a possibilidade de novos perfis através do uso de exibidores de mídia de terceiros integrados ao ambiente Ginga-NCL. Esses exibidores de terceiros devem ser integrados ao ambiente Ginga-NCL por meio adaptadores, como é ilustrado Figura 10; e a H.762 (ITU, 2011), que descreve o *framework* LIME derivado do BML (ARIB, 2004), constituído por subconjuntos das especificações *XHTML*, *CSS*, *DOM* e *ECMAScript*, respectivamente chamados de *LIME-HTML*, *LIME-CSS*, *LIME-DOM* e *LIME-SCRIPT*.

**Figura 10: Integrando exibidores de mídia de no ambiente Ginga-NCL.**



Fonte: (ITU, 2009b).

#### 2.2.4. Aplicações Ginga-NCL com segunda tela

A evolução de tecnologias de computação em rede juntamente com a popularização de dispositivos portáteis propiciaram cenários que integram recursos presentes nesses dispositivos, permitindo que múltiplos dispositivos sejam interconectados e cooperem dentro da lógica de uma aplicação. Dentre essas aplicações estão a TV Digital integrada com dispositivos de segunda tela.

As aplicações segunda tela em TV Digital se caracterizam por realizar uma apresentação multimídia distribuída entre o receptor de TV Digital e dispositivos portáteis, como celulares, *tablets*, entre outros. Exemplos desse cenário são aplicações de compartilhamento de mídias através de *UPnP*, navegação, controle remoto virtual, conteúdo adicional, individualização de conteúdo, acessibilidade, libras, áudio-descrição, entre outros.

Nas especificações de *middleware* terrestre internacionais existem propostas de como oferecer suporte à execução desse tipo de aplicação. No *middleware* MHP propostas como a de (FERN, 2006) visam integrar aplicações com o framework *OSGi*. Essas propostas permitem que os *Xlets MHP* possam usar os serviços *OSGi* associados a dispositivos registrados dinamicamente. Na especificação do *middleware* japonês ARIB, Takechi et al. (2011) definem uma API através do pacote `jp.or.arib.tv.peripheral` que viabiliza a obtenção de recursos a partir de dispositivos conectados a um receptor de TV. Entretanto com suporte limitado para tecnologias *UPnP*, Bluetooth e conexões de dispositivos USB.

Na versão inicial do ambiente Ginga-J (DE SOUZA FILHO, et al. 2007) foi um API proposta (SILVA et al. 2008) para integração com dispositivos para a troca de vários formatos de mídia, como, áudio, vídeo, teclas pressionada, etc. Esta especificação não defini protocolos de como a comunicação é realizada entre os dispositivos.

No do ambiente Ginga-NCL é possível associar uma entidade de *region* com um dispositivo de segunda tela através do atributo *device*. Uma mídia associada a essa *region* será apresentada na classe de dispositivo definida. Logo esta especificação também não se define quais os protocolos que serão utilizados para a comunicação entre os dispositivos.

Costa et al. (2009) propõe dois tipos de classes de dispositivos de *segunda tela* para exibição de mídias em aplicações NCL. A primeira, chamada de classe passiva, o mesmo conteúdo é apresentado nos dispositivos registrados na classe. O segundo tipo, chamado de classe ativa, permite que os dispositivos registrados na classe controlem a apresentação de um documento NCL independente.

Constata-se que nenhuma das abordagens citadas realiza uma definição de como deve ser a comunicação do receptor com os dispositivos, delegando aos fabricantes dos receptores implementar seus próprios mecanismos de comunicação. Esta lacuna foi preenchida pelo trabalho de (BATISTA et al., 2010) no ambiente Ginga-NCL.

O trabalho de (BATISTA et al., 2010) propõe um extensão ao mecanismo de definição de classes de dispositivos existente na especificação Ginga-NCL. Nele o autor associa uma *region* de dispositivo de segunda tela a um arquivo de descrição *UAProf* referenciado por uma *tag META*. Através dessa descrição o ambiente Ginga-NCL consegue realizar a descoberta e estabelecer comunicação com dispositivos de *segunda tela*.

A extensão do formato do *UAProf* contempla o acréscimo de alguns elementos específicos do modelo de múltiplos dispositivos proposto para o Ginga-NCL com o objetivo de incorporar outras semânticas e capacidades de dispositivo. A definição das classes utiliza as extensões aos elementos *HardwarePlatform*, *SoftwarePlatform* e *NetworkCharacteristics* do *UAProf*, e uma nova classe chamada *DeviceGroup*. As tabelas a seguir apresentam as extensões aos elementos *UAProf* e a definição da classe *DeviceGroup*.

**Tabela 2: Definição da classe *DeviceGroup*.**

<b>Classe <i>UAProf DeviceGroup</i></b>	
<i>Attribute</i>	<i>Type</i>
<i>gncl:maxDevices</i>	<i>Number</i>
<i>gncl:minDevices</i>	<i>Number</i>
<i>gncl:groupType</i>	“passive”   “active”
<i>gncl:HardwareRequirement</i>	<i>prf:HardwarePlatform</i>
<i>gncl:SoftwareRequirements</i>	<i>prf:SoftwarePlatform</i>
<i>gncl:NetworkRequirements</i>	<i>prf:NetworkCharacteristics</i>

Fonte: Adaptado de (BATISTA et al., 2010)

**Tabela 3: Atributos adicionados às classes *UAProf*.**

<b>Classe <i>SoftwarePlatform</i></b>		
<b>Atributo</b>	<b>tipo</b>	<b>Exemplos</b>
<i>gncl:requiredNCLProfile</i>	<i>rdf:Bag</i>	“NCL3.0/EDTV”
<i>gncl:deviceServices</i>	<i>rdf:Bag</i>	“UPnP_MSCP”, “ActiveClas
<i>gncl:softwareParams</i>	<i>rdf:Bag</i>	“KEY=123456”
<b>Classe <i>NetworkCharacteristics</i></b>		
<b>Atributo</b>	<b>tipo</b>	<b>Exemplos</b>

<i>gncl:registrationService</i>	<i>Literal</i>	“UPnP”
<i>gncl:requiredProtocols</i>	<i>rdf:Bag</i>	“HTTP”, “RTP”
<i>gncl:networkParams</i>	<i>rdf:Bag</i>	“IP=10.0.1.1”

Fonte: Adaptado de (BATISTA et al. 2010)

Além do mecanismo extensão *UAProf*, (BATISTA et al., 2010) propõe o uso de uma API Lua chamada *LuaTV* (BRANDÃO et al., 2010), que tem o intuito de oferecer suporte através de scripts Lua à criação dinâmica de classes e oferecer outras funcionalidades relacionadas ao modelo de integração de dispositivos do Ginga-NCL.

A API *LuaTV* é composta por dois módulos, chamados de *devicemanager* e *deviceservice*. Uma nova classe de eventos NCLua foi criada, possuindo como identificador a *string device*. O módulo *devicemanager* é responsável pelo registro de classes através de informações RDF serializadas (função *devicemanager:registerClass*), pelo acesso aos metadados das classes de dispositivos registradas (função *devicemanager:classMetadata*) e a quantidade de dispositivos de uma classe (função *devicemanager:deviceClassSize*).

### 3. Trabalhos Relacionados

Este capítulo apresenta trabalhos encontrados na literatura que apresentam semelhanças com a solução proposta nesta dissertação. Dois tipos de trabalhos foram elencados para comparações: os que lidam com o desenvolvimento de repositórios de aplicações (*Application Store*) e os que possuem suporte à execução de aplicações de segunda tela.

*Yahoo Connected TV*<sup>1</sup> é o nome da plataforma de aplicações interativas para *Broadband TV* desenvolvida pela *Yahoo* em parceria com fabricantes de eletrônicos de consumo. Esse ambiente é focado no uso de *widgets* com conteúdos e serviços sem relacionamento com programação assistida em broadcast. *Yahoo Connected TV* visa permitir aos usuários de TV o uso de serviços populares da *Web*, em especial os desenvolvidos pela própria *Yahoo*, como *Yahoo Finance*<sup>2</sup>, *Yahoo Sports*<sup>3</sup>, *Flicker*<sup>4</sup>, entre outros.

O desenvolvimento das aplicações para a plataforma *YahooTV* é baseado no modelo *Widget KONTx Framework*<sup>5</sup>, que utiliza as tecnologias *Web* como *HTML* e *JavaScript*. A plataforma da *Yahoo* possui uma API *Javascript* para criação de *widget* gráficos, acesso a serviços decodificação de mídia e a metadados dos serviços.

A plataforma *Yahoo Connected TV* inclui uma API (YAHOO, 2011) para integração de dispositivos, que permite o desenvolvimento de aplicações com uso de dispositivos de segunda tela. É necessária a instalação de uma aplicação da *Yahoo*<sup>6</sup> nos dispositivos a qual realiza a descoberta de serviços com a biblioteca *mDNSResponde* (pertencente ao protocolo *Boujour*<sup>7</sup>),

---

<sup>1</sup> <http://connectedtv.yahoo.com>. Acessado em: agosto de 2012

<sup>2</sup> <http://finance.yahoo.com>. Acessado em: agosto de 2012

<sup>3</sup> <http://sports.yahoo.com>. Acessado em: agosto de 2012

<sup>4</sup> <http://www.flickr.com>

<sup>5</sup> <http://developer.yahoo.com/connectedtv/kontxapi>. Acessado em: agosto de 2012

<sup>6</sup> <http://play.google.com/store/apps/details?id=com.yahoo.connectedtv.yremote>. Acessado em: agosto de 2012

<sup>7</sup> <http://www.apple.com/br/support/bonjour>. Acessado em: agosto de 2012

criando assim uma configuração que dificulta a interoperabilidade com outras plataformas para aplicações multidispositivo.

Como ilustrado na Figura 11, a API limita os recursos das aplicações de segunda tela a mensagens que carregam informações de interação, como teclas de navegação e dados arbitrários em formato de texto.

**Figura 11: Telas da aplicação de controle de segunda tela do *Yahoo Connected TV*.**



Fonte: (YAHOO, 2011).

O Google TV<sup>1</sup> é uma plataforma de *software* para *Broadband TV* desenvolvida pelo Google em parceria com fabricantes de eletrônicos de consumo, sendo baseada no sistema operacional *Android*<sup>2</sup>. De modo semelhante ao *Yahoo Connected TV*, as aplicações dessa plataforma utilizam uma API proprietária (GOOGLE, 2012), que possibilita a utilização de dispositivos portáteis como segunda tela, desde que esses possuam a aplicação *Android* chamada Google TV Remote<sup>3</sup>. Novamente é criada uma configuração que dificulta a interoperabilidade com outras plataformas para aplicações multidispositivo.

O Google TV Remote, Ilustrado na Figura 12, permite que aplicações Google TV utilizem um celular ou *tablet* com um controle remoto avançado, com controle de cursor ou teclado da TV. A comunicação entre os dispositivos portáteis com o receptor do Google TV é realizado

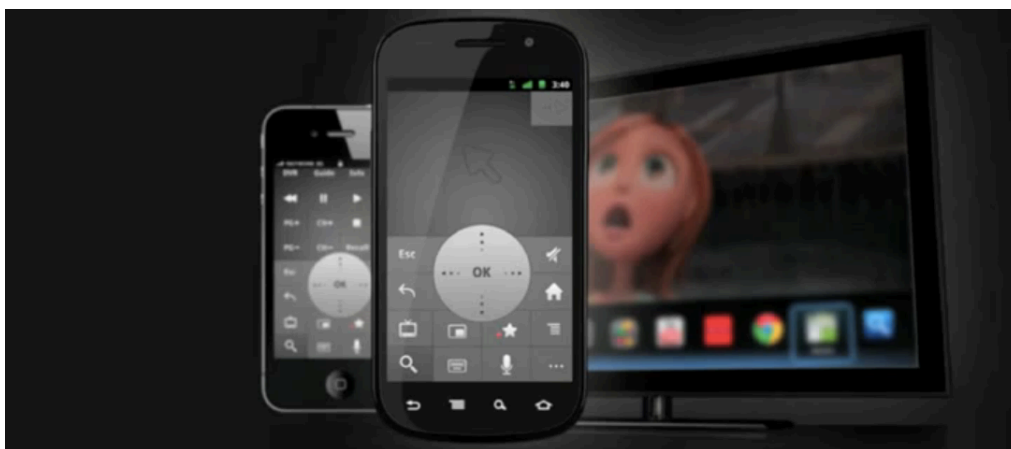
<sup>1</sup> <http://www.google.com/tv>. Acessado em: agosto de 2012

<sup>2</sup> <http://www.android.com>. Acessado em: agosto de 2012

<sup>3</sup> <http://play.google.com/store/apps/details?id=com.google.android.apps.tvremote>. Acessado em: agosto de 2012

através do protocolo Anymote<sup>1</sup>. O protocolo descreve mensagens específicas que encapsular informações de descrição dos dispositivos, códigos de teclas e de seleção via cursor, além de *strings* arbitrárias que encapsulam informações referentes à ativação de exibidores de mídia (páginas Web, vídeos etc.).

**Figura 12: Tela da aplicação com uso de segunda tela do GoogleTV.**



Fonte: (GOOGLE, 2012).

O *StickerCenter*<sup>2</sup> é a loja de aplicações no modelo de *Broadband TV* que é instalada em conjunto com a implementação do *middleware Ginga* da empresa TOTVS chamada *AstroTV*<sup>3</sup>. A loja permite a recuperação execução de aplicações interativas na TV, chamadas de *Stickers*. Os *Stickers* são aplicações Lua, que não são totalmente compatíveis com a especificação Ginga, utilizando uma biblioteca Lua (TOTVS. 2010) de *widgtes* gráficos proprietária.

Recentemente (BOQUIMPANI, 2012) foi apresentado um modelo de interatividade em segunda tela para o *StickerCenter*. Esse modelo, ilustrado na Figura 13, tem a estratégia de sinalizar para um aplicação instalada em um dispositivo *Android*<sup>4</sup> para acessar uma *URL* da *Web* relacionado à lógica da aplicação executada no receptor.

O trabalho de LI et al. (2008) aborda o problema da necessidade de ligar a descoberta e a descrição de dispositivos dentro de um contexto de execução a fim enriquecer a experiência multimídia. Nele o autor salienta que esforços para construir representações OWL de padrões de

<sup>1</sup> <http://code.google.com/p/anymote-protocol>. Acessado em: agosto de 2012

<sup>2</sup> <http://www.stickercenter.com.br>. Acessado em: agosto de 2012

<sup>3</sup> <http://www.astrodevnet.com/AstroDevNet/home.html>. Acessado em: agosto de 2012

<sup>4</sup> <http://play.google.com/store/apps/details?id=br.com.totvs.ecmbox>. Acessado em: agosto de 2012

multimídia como MPEG-7(ISO/IEC, 2000) acabam se tornando pouco expressivas, argumentando ser necessária uma conversão manual e propondo uma descrição compatível com a DIA MPEG-21(ISO/IEC, 2004).

**Figura 13: Sinalização do *Sticker Center* a dispositivos de segunda tela.**



Fonte: (BOQUIMPANI, 2012)

A contribuição desse trabalho consiste no mapeamento de descrições de dispositivos utilizando XSTL<sup>1</sup>. Esse trabalho usa descrição *UPnP* como um mapeamento possível. Com essa abordagem o autor fornece uma solução que pretende preencher a lacuna entre os mecanismos de descrição não semânticos de dispositivos.

O trabalho de TOGIAS et al. (2010) apresenta uma representação de base ontológica para dispositivos *UPnP* e serviços que proporcionam uma ligação semântica entre conceitos humanos, por exemplo, descrições como “*reading*” e “*study*”, e a descrição do *software* baseada em descritores XML obtidos do *UPnP*.

Como pode ser observado na Figura 14, a “Physical Device Ontology” possui tanto entidades de semântica humana como *Location*, *User* e *usefullFor*, quanto entidades de descrição dos dispositivos como *upnpdevice* e *upnpservice*. Ambientes com dispositivos *UPnP* que estão equipados com as instâncias dessa descrição permitem realizar adaptação ao contexto realizando a seleção do dispositivo tomando por base atividades humanas.

**Figura 14: Descrição ontológica da Physical Device.**

<sup>1</sup> <http://www.w3.org/TR/xsl>. Acessado em: agosto de 2012



Fonte: (TOGIAS et al., 2010)

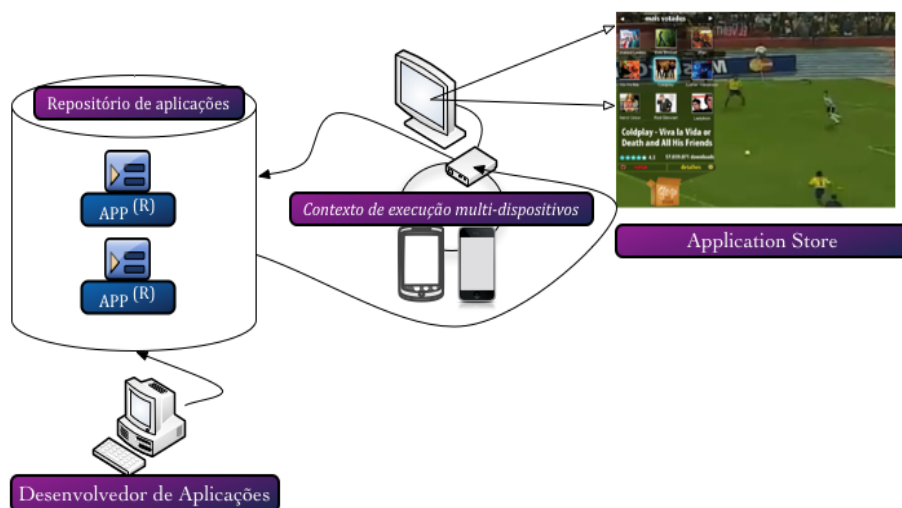
## 4. Arquitetura de *software* da Solução

Como apresentado na seção de motivação, esta dissertação tem o intuito de abordar a integração de dois cenários de uso de interatividade na TV Digital: o das aplicações em receptores de TV Digital com acesso a banda larga, e o de aplicações de TV Digital interativa que utilizam o conceito de segunda tela. Nesta dissertação é discutida uma proposta de arquitetura para uma *Application Store* destinada a receptores *Broadband TV*. O diferencial do serviço proposto é o tratamento da heterogeneidade dos múltiplos dispositivos envolvidos na execução de uma aplicação. A seguir neste capítulo, são apresentados os requisitos desse sistema através de um cenários de uso e a descrição formal de sua arquitetura.

### 4.1. Cenário de uso e requisitos de alto nível

Para apresentar o cenário de uso, é necessária uma visão geral de um sistema que lhe dê suporte, a qual é ilustrado na Figura 15.

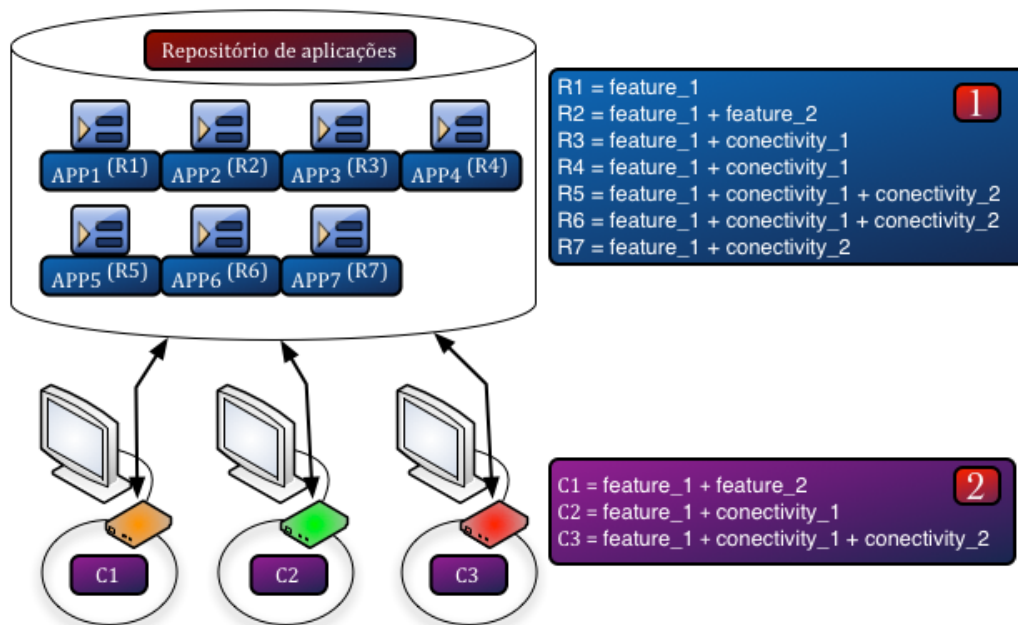
Figura 15: Visão geral do sistema



Na visão geral ressaltamos a presença de quatro atores: o desenvolvedor de aplicações, o repositório de aplicações, o contexto de execução (dispositivos usados para execução) do receptor de TV que acessa o repositório; e o serviço de *Application Store* que recupera e executa aplicações compatíveis com o contexto de execução.

Cada aplicação armazenada no repositório está associada a um conjunto de requisitos para sua execução, sejam elas *stand-alone* ou uma que utiliza conectividade com dispositivos de segunda tela. Cada contexto possui um conjunto de recursos multidispositivos. A Figura 16 ilustra uma instância do cenário de uso onde o repositório possui sete aplicações (Figura 16-1) e é acessado por três contextos de usuários (Figura 16-2).

**Figura 16: Instância de cenário de uso com parâmetros suas aplicações e contextos de execução multidispositivos**



Existem duas aplicações que não utilizam o conceito de segunda tela (*APP1* e *APP2*), e as demais utilizam esse recurso. Todas as aplicações requerem as funcionalidades de receptor *feature\_1* e a *APP2* requer a funcionalidade *feature\_2*. Das aplicações que usam segunda tela, quatro requerem a *conectividade\_1* (*APP2*, *APP3*, *APP4* e *APP5*) e três a *conectividade\_2* (*APP5*, *APP6* e *APP7*).

Todos os contextos de execução oferecem a funcionalidade *feature\_1*. O contexto **C1** que oferece a característica *feature\_2* está apto a receber as aplicações *stand-alone* (*APP1* e *APP2*). O contexto **C2**, que possui a característica *conectividade\_1*, está habilitado a receber as aplicações

*APP1*, *APP3* e *APP4*. E o contexto **C3**, que possui as características *connectivity\_1* e *connectivity\_2*, pode receber todas as aplicações, menos a *APP2*, dado que ela requisita a funcionalidade *feature\_2*.

A partir desse cenário é possível levantar os seguintes requisitos de alto nível que servem como base para a especificação de requisitos funcionais e não funcionais da arquitetura do sistema proposto:

**(I) Interoperabilidade de aplicações multidispositivos:** As aplicações armazenadas no repositório devem ser compatíveis com um formato interoperável de aplicação multidispositivo, que possa ser executado em diferentes contextos de dispositivos.

**(II) Recuperação de aplicações dado o contexto de execução multidispositivos:** A *Application Store* deve ser capaz de recuperar a descrição do contexto de execução multidispositivos em que está presente. Isso permite a recuperação de aplicações compatíveis advindas do repositório.

**(III) Portabilidade da *Application Store*:** A *Application Store* deve ser capaz de oferecer a recuperação e a execução em diferentes contextos de execução multidispositivo.

**(IV) Serviço de repositório de aplicações:** O repositório de aplicações deve armazenar as aplicações e suas descrições de requisitos. Isso permite que o desenvolvedor realize a associação dos requisitos da aplicação, viabilizando um serviço de recuperação das aplicações compatíveis com contexto de execução de cada usuário.

## 4.2. Subsistemas e requisitos de software

Com o intuito de implementar os requisitos de alto nível apresentados, é importante ressaltar que a implementação de referência do middleware Ginga e a especificação H.761 influenciaram muito o desenvolvimento desta dissertação. O Ginga fornece a infraestrutura necessária para suprir o primeiro dos requisitos de alto nível propostos e provê recursos para a implementação do segundo e terceiro. Logo, a implementação de referência do Ginga será utilizada para dar suporte à execução da aplicação proposta no cenário de uso.

A *Application Store*, a partir de agora chamada de *GingaStore*, constitui uma aplicação Ginga-NCL que é responsável por recuperar e executar outras aplicações Ginga-NCL. Ela é capaz de recuperar uma descrição do contexto do ambiente Ginga-NCL em termos de funcionalidades de execução *stand-alone* e serviços de conectividade com dispositivos de

segunda tela. Essa descrição permite ao *GingaStore* acessar o repositório e recuperar aplicações considerando seu contexto de dispositivos.

O repositório de aplicações, a partir de agora chamado *GingaSpace*, consiste na entidade que armazena aplicações NCL e oferece um serviço para recuperação das mesmas. Uma aplicação disponível no repositório poderá ser recuperada e executada em receptores por intermédio do cliente *GingaStore*, a ser executada pelos receptores de TV. Esse serviço de recuperação deverá filtrar as aplicações do repositório a partir dos recursos acessíveis pela *GingaStore*.

As Tabelas 4 e 5 listam os requisitos funcionais e não funcionais dos subsistemas. Os requisitos funcionais do sistema são classificados em evidentes e ocultos ao usuário final. Os requisitos evidentes são percebidos pela interface com usuário, enquanto que os ocultos são tratados pelos subsistema sem o conhecimento explícito do mesmo.

**Tabela 4: Requisitos relacionados ao *GingaStore***

Ref. #	Nome do requisito	Categoria	Requisito de alto nível associado
<b>Requisitos Funcionais do <i>GingaStore</i></b>			
F1	Listar aplicações disponíveis	Evidente	(I)
F2	Exibir descrição de Aplicação	Evidente	(I)
F3	Instalar Aplicação	Evidente	(I)
F4	Executar Aplicações de TV	Evidente	(I)
F5	Executar Aplicações de TV com segunda tela	Evidente	(I)
F6	Identificação de conectividade de dispositivos de	Evidente	(II)
F7	Recuperação de contexto multidispositivos	Oculto	(II)
<b>Requisitos Não Funcionais do <i>GingaStore</i></b>			
NF1	Portabilidade de execução do <i>GingaSpace</i> em ambientes	Oculto	(III)
NF2	Interoperabilidade de aplicações pelo H.761	Oculto	(I)
NF3	Interoperabilidade de serviços de segunda tela	Oculto	(I)
NF4	Interface de loja deve evidenciar as aplicações disponíveis na loja e as instaladas no receptor	Evidente	(I), (IV)
NF5	Interface de loja não deve interferir no conteúdo	Evidente	(I), (IV)
NF6	Interface da loja deve evidenciar a conectividade de dispositivos na execução de aplicações com segunda tela	Evidente	(I), (II)
NF7	Interface da loja deve listar as aplicações por categorias do tipo de serviço oferecido	Evidente	(I)

**Tabela 5: Requisitos relacionados repositório *GingaSpace***

Ref. #	Nome do requisito	Categoria	Requisito de alto nível associado
<b>Requisitos Funcionais do <i>GingaSpace</i></b>			
F8	<i>GingaStore</i> recuperar lista de aplicações disponíveis	Evidente	(IV)
F9	<i>GingaStore</i> recupera meta-dados de armazenamento de	Oculto	(IV)
F10	Armazenar aplicações	Oculto	(IV)
F11	Armazenar meta-dados de aplicações	Oculto	(II),(IV)
F12	Cadastrar desenvolvedor	Evidente	(IV)
F13	Cadastra aplicações	Evidente	(II),(IV)
F14	Cadastra aplicações com segunda tela	Evidente	(II),(IV)
F15	Identificação de conectividade de dispositivos de	Evidente	(IV)
F16	Recuperação de contexto multidispositivos	Oculto	(IV)
<b>Requisitos Não Funcionais do repositório do <i>GingaSpace</i></b>			
NF7	Controle do acesso de clientes <i>GingaSpace</i>	Oculto	(IV)
NF8	Controle do acesso dos desenvolvedores	Evidente	(IV)
NF9	Interfaces de recuperação portátil para clientes de TV	Oculto	(III)

As próximas seções descrevem a arquitetura do sistema utilizando o esquema formal de descrição para arquiteturas de *software* proposto por Kruchten (1995) chamado “4+1”. Esse esquema apresenta as características da arquitetura em diagramas UML<sup>1</sup>, definindo quatro visões (implantação, processos, lógica e implementação) coordenadas por uma quinta visão de casos de uso. Essa descrição permite evidenciar as contribuições da arquitetura proposta, prover mecanismos formais de validação da arquitetura e permitir a replicabilidade do trabalho aqui proposto.

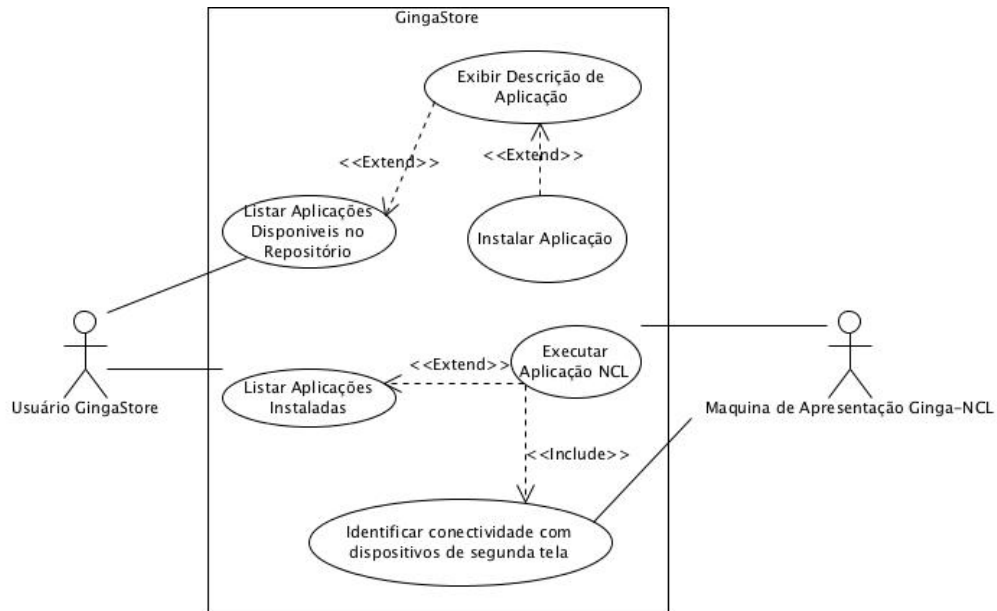
### 4.3. Visão de casos de uso

A visão de casos de uso resume o comportamento do sistema a partir de diagramas UML de casos de uso. Essa visão é considerada o ponto de partida e também de consolidação das outras visões.

A Figura 17 apresenta o diagrama de casos de uso do *GingaStore*, que permite a compreensão do comportamento externo subsistema e uma visão geral das funções e serviços fornecidos aos usuários.

<sup>1</sup> Booch, Grady, James Rumbaugh, and Ivar Jacobson. 1998. The Unified Modeling Language User Guide. Addison Wesley.

**Figura 17: Diagrama de casos de uso do subsistema *GingaStore***



Nesse diagrama é possível identificar dois atores relevantes: o usuário do receptor de TV que lista, instala e executa aplicações no *GingaStore*, e o ambiente de apresentação *Ginga-NCL*, que permite a execução das aplicações.

A Figura 18 apresenta o diagrama de casos de uso do repositório *GingaSpace*, que permite a compreensão do comportamento externo do subsistema e uma visão geral das funções e serviços fornecidos aos usuários. Os atores envolvidos com o repositório *GingaSpace* e o *GingaStore* que interage de modo a adquirir e executar aplicações existentes armazenadas e os desenvolvedores destas.

**Figura 18: Diagrama de casos de uso do subsistema *GingaSpace***



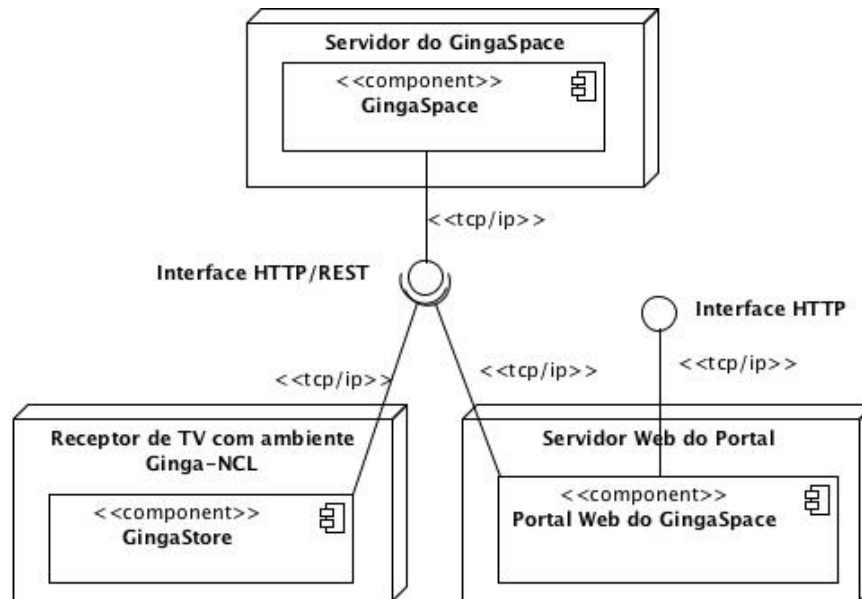
Nesse diagrama é possível identificar três atores relevantes: O *GingaStore*, que acessa as aplicações e meta-dados do seu repositório; um desenvolvedor anônimo que não possui cadastro no repositório, se limitando a conhecer as aplicações do repositório e acessar informações sobre desenvolvimento; e o desenvolvedor registrado no repositório que cadastra aplicações.

#### 4.4. Visão de implantação

A visão de implantação descreve os componentes de *software* do sistema os posicionando em nós físicos de computação, permitindo avaliar requisitos não funcionais como desempenho, disponibilidade, confiabilidade e escalabilidade.

O diagrama de implantação apresentado na Figura 19 mostra a arquitetura distribuída e a configuração física dos componentes de *software* do sistema. É possível, então, visualizar quais nós são necessários para executar o sistema e como eles estão conectados. Nesse diagrama é possível identificar as seguintes entidades computacionais distintas: o receptor de TV com o *Ginga-NCL*, onde é executada a aplicação *GingaStore*; o servidor onde é executado o repositório *GingaSpace*; e o servidor *Web* que contém um componente de Portal Web para gestão dos acesso de desenvolvedores e cadastro de suas aplicações no repositório.

**Figura 19: Diagrama de implantação do sistema**



Ressalta-se a distinção de acesso dos atores envolvidos no *GingaSpace*: desenvolvedores de aplicações e a aplicação *GingaStore*. O repositório *GingaSpace* se apresenta através de dois componentes em máquinas distintas. Essa distinção possui três implicações: oferecer uma interface com usuário específica para desenvolvedores; possibilitar a escalabilidade do Servidor do *GingaSpace*, considerando a quantidade de usuários de um sistema de TV Digital; e definir uma interface de serviço de recuperação que possibilite as instancias de *GingaStore* serem interoperáveis e com um tráfego de dados na rede eficiente, considerando as limitações de configurações dos receptores de TV.

Dois modelos de arquitetura para serviços *Web* foram considerados para definir a interface de acesso do componente repositório *GingaSpace*: o protocolo baseado em XML SOAP<sup>1</sup> e o estilo modelo REST.

Duas características motivaram a escolha pelo modelo REST. Primeiro a compatibilidade com requisições HTTP<sup>2</sup>, que permite a portabilidade dos clientes do repositório. E segundo pelo melhor desempenho dos serviços REST (que possibilita troca mensagens no formato *JSON* -

<sup>1</sup> <http://www.w3.org/TR/soap/>. Acessado em: agosto de 2012

<sup>2</sup> <http://www.w3.org/Protocols/rfc2616/rfc2616.html>. Acessado em: agosto de 2012

*Java Script Object Notation*<sup>1</sup>) em contraposto ao *XML SOAP*, considerado “*Big Web Services*” (PAUTASSO, 2008).

Dentre as soluções de autenticação e acesso ao repositório *GingaSpace* consideradas, o protocolo aberto *OAuth*<sup>2</sup> foi o que melhor atendeu aos requisitos. Primeiramente se deve sua atuação sinérgica com a arquitetura *REST* e *HTTP*, como exemplificado em serviços como *Twitter*<sup>3</sup> e *Facebook*<sup>4</sup>. Permitindo, em sua versão *OAuth2*<sup>5</sup>, que as requisições sejam feitas apenas em *HTTP*, evitando uma dependência a *SSL*(STALLINGS, 2007) no *GingaStore*.

O *OAuth* realiza o controle de acesso a recursos através da distribuição de chaves para aplicações com escopo determinado, chamadas *OAuth token*. Logo, enquanto as chave *OAuth token* nos clientes escopo apenas de recuperação, o Portal *Web* possui chave para modificações ao *GingaStore*.

## 4.5. Visão de processos

A visão de processos documenta a colaboração entre os componentes do sistema dentro dos nós físicos de computação, evidenciando o processamento de tarefas no sistema. Ela é usualmente descrita por diagramas UML de atividades e colaboração.

O digrama de colaboração exibido na Figura 20 evidencia a colaboração das entidades do sistema. O desenvolvedor de aplicações, após realizar autoria da aplicação, tem a responsabilidade de se registrar e cadastrar a aplicação no Portal *Web* do *GingaSpace* (mensagens 1, 2 e 3). Nesse momento ele deve descrever quais são os requisitos dos dispositivos necessários para a correta execução da aplicação e os registrar no repositório de aplicações. Após esse processo, o Portal *Web* se comunica com o *GingaSpace* para armazenar as aplicações e os seus meta-dados de descrição (mensagens 4).

Para recuperar aplicações do *GingaSpace*, cada instância da aplicação *GingaStore* deve realizar sua autenticação junto ao repositório (mensagem 5). Em seguida, para realizar a

---

<sup>1</sup> <http://www.json.org>. Acessado em: agosto de 2012

<sup>2</sup> <http://oauth.net/>. Acessado em: agosto de 2012

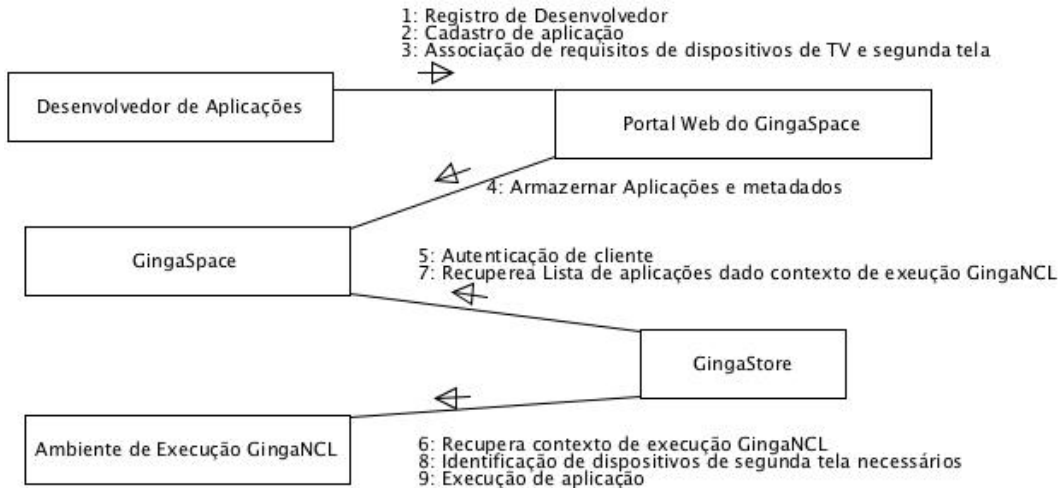
<sup>3</sup> <http://dev.twitter.com/docs/auth/oauth>. Acessado em: agosto de 2012

<sup>4</sup> <http://developers.facebook.com/docs/authentication>. Acessado em: agosto de 2012

<sup>5</sup> <http://oauth.net/2/>. Acessado em: agosto de 2012

requisição de uma lista de aplicações (mensagens 7), o cliente deve recuperar o contexto do execução do Ginga-NCL em que está inserido (mensagens 6).

**Figura 20: Diagrama de colaboração entre atores do sistema**



A partir da lista de aplicações compatíveis com um contexto de execução em particular, a *Application Store* realiza a apresentação das aplicações para o usuário. Após o usuário escolher uma aplicação, o *GingaStore* deve verificar o estado relativo à conectividade dos dispositivos de segunda tela requisitados junto ao receptor com o Ginga-NCL (mensagem 8). Por fim, pode-se invocar a execução da aplicação escolhida no ambiente Ginga-NCL (mensagem 9).

## 4.6. Visão lógica

A visão lógica tem o intuito de descrever elementos comportamentais e funcionalidades da arquitetura. Nessa dissertação, essa visão tem o foco de: detalhar a descrição de contexto multidispositivos do ambiente Ginga-NCL utilizada pelo repositório *GingaSpace*; e descrever o comportamento da aplicação *GingaStore*.

Apresentadas na sessão 2.1, diferentes descrições de dispositivos são utilizados descrever contexto de dispositivos. No entanto, foi a descrição *UAProf* estendida de Batista et al. (2010) que ofereceu a base para propor nossa descrição multidispositivos do ambiente Ginga-NCL. Essa descrição deve considerar a heterogeneidade de funcionalidades do receptor e de conectividades com dispositivos de segunda tela apresentadas no cenário uso.

As funcionalidades de receptor são caracterizadas no Ginga-NCL pelos tipos de mídia que um receptor é capaz de controlar. Segundo a especificação H.761, os receptores podem integrar

exibidores de mídias arbitrários, criando diferentes perfis de ambiente de execução para o Ginga-NCL. A proposta desta dissertação é que o *GingaStore* realize a **descoberta dos exibidores suportados pelo ambiente Ginga-NCL**.

As conectividades com dispositivos são caracterizadas no Ginga-NCL pelas classes de dispositivos suportadas. A solução de Batista et al. (2010) estende as especificações de classes originais do Ginga-NCL (Costa et al., 2009), deixando o Ginga-NCL não restritivo às classes de dispositivos suportadas. Essa solução defini, através de documentos *UAPProf*, como deve o ser a comunicação do receptor com os dispositivos de segunda tela heterogêneos. A proposta desta dissertação é que o *GingaStore* realize a **descoberta das classes de dispositivos suportados pelo ambiente Ginga-NCL considerando a descrição de Batista et al. (2010)**.

Logo para descrever um ambiente multidispositivos Ginga-NCL a descrição deve abordar os seguintes requisitos do receptor: **(A)** os perfis NCL necessários; **(B)** os exibidores de mídia não presentes no perfil; e **(C)** serviços de conectividade com classes de dispositivos de segunda tela.

Com o intuito de abordar esses requisitos esta dissertação propõe uma descrição que atua de modo semelhante à entidade *DeviceGroup* (BATISTA et al., 2010), mas caracterizando o ambiente Ginga-NCL multidispositivos do receptor. Ela trata os requisitos citados utilizando respectivamente os seguintes elementos *UAPProf* do componente *prf:SoftwarePlatform*: a extensão *gncl:requiredNCLProfiles* de (BATISTA et al., 2010) para definir os perfis NCL **(A)**; o próprio elemento *UAPProf* para mime-types<sup>1</sup>, chamado *prf:CcppAccept*, para definir para os exibidores de mídia **(B)**; e a extensão *gncl:deviceService* de (BATISTA et al., 2010) para os serviços de conectividade **(C)**. A Figura 21 ilustra duas descrições RDF de ambientes multidispositivo Ginga-NCL de receptores.

---

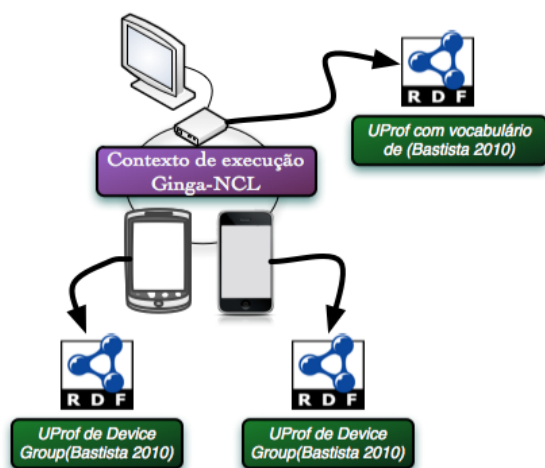
<sup>1</sup> <http://www.iana.org/assignments/media-types/index.html>. Acessado em: agosto de 2012.

**Figura 21: Um descrição de contexto de execução Ginga-NCL de perfil EDTV e suporte a mídias *HTML5*, e outro de perfil EDTV conectividade *UPnP***

<pre> 6 &lt;rdf:Description rdf:ID="DeviceProfile"&gt; 7   &lt;gnc1:component&gt; 8     &lt;gnc1:Description rdf:ID="SoftwarePlatform"&gt; 9       &lt;rdf:type rdf:resource=prf:SoftwarePlatform/&gt; 10      &lt;gnc1:supportedNCLProfiles&gt; 11        &lt;rdf:Bag&gt; 12          &lt;rdf:li&gt;NCL3.0/EDTV&lt;/rdf:li&gt; 13        &lt;/rdf:Bag&gt; 14      &lt;/gnc1:supportedNCLProfiles&gt; 15      &lt;prf:CcppAccept&gt; 16        &lt;rdf:Bag&gt; 17          &lt;rdf:li&gt;application/xhtml+xml&lt;/rdf:li&gt; 18        &lt;/rdf:Bag&gt; 19      &lt;/prf:CcppAccept&gt; 20    &lt;/rdf:Description&gt; 21  &lt;/prf:component&gt; 22 &lt;/rdf:Description&gt; 23 &lt;/rdf:RDF&gt; </pre>	<pre> 6 &lt;rdf:Description rdf:ID="DeviceProfile"&gt; 7   &lt;gnc1:Description rdf:ID="SoftwarePlatform"&gt; 8     &lt;gnc1:supportedNCLProfiles&gt; 9       &lt;rdf:Bag&gt; 10        &lt;rdf:li&gt;NCL3.0/EDTV&lt;/rdf:li&gt; 11      &lt;/rdf:Bag&gt; 12    &lt;/gnc1:supportedNCLProfiles&gt; 13    &lt;gnc1:deviceServices&gt; 14      &lt;rdf:Bag&gt; 15        &lt;rdf:li&gt;UPnP_MSCP&lt;/rdf:li&gt; 16      &lt;/rdf:Bag&gt; 17    &lt;/gnc1:deviceServices&gt; 18  &lt;/rdf:Description&gt; 19 &lt;/prf:component&gt; 20 &lt;/rdf:Description&gt; 21 &lt;/rdf:RDF&gt; </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

É importante ressaltar que a descrição proposta não tem o intuito de substituir a descrição de (BATISTA et al., 2010), mas trabalhar em sinergia. Enquanto a descrição proposta tem o intuito de descrever o ambiente Ginga-NCL e suas conectividades com classes de dispositivos, Batista et al. (2010) descrevem classes de dispositivos e o comportamento do ambiente Ginga-NCL para apresentação distribuída. Como lustrado na Figura 22, cada aplicação no repositório deve possuir *RDFs* associados para descrever o ambiente de execução Ginga-NCL e as classes de dispositivos que utiliza.

**Figura 22: Visão geral da descrição de requisitos de aplicação no *GingaSpace***



O requisito de alto nível (I) de interoperabilidade de aplicações é atendido graças ao uso do ambiente Ginga-NCL e sua não restrição de mídias e dispositivos de segunda tela. Logo a interoperabilidade de aplicações no *GingaStore* é gerida pelo próprio Ginga-NCL e seus exibidores de mídia. É baseada nessa Interoperabilidade do Ginga-NCL é possível alcançar o

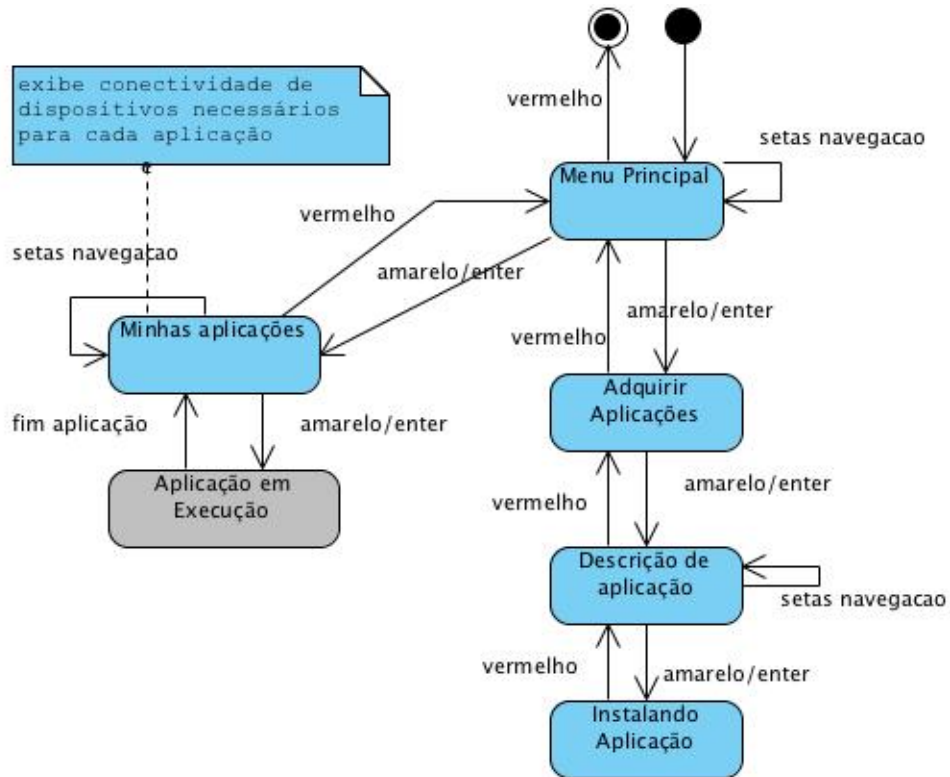
requisito de alto nível (III) de portabilidade do *GingaStore*, a medida de ser proposto ela seja baseada apenas no ambiente Ginga-NCL.

A principal dificuldade de implementar o *GingaStore* utilizando o ambiente Ginga-NCL se configura em como carregar integralmente e dinamicamente outras aplicações para serem executadas. Para contornar esta dificuldade foi utilizado o conceito de aplicações NCL dinamicamente reconfiguráveis proposto por de Sousa Junior (2012). Essas aplicações NCL são compostas por mídias pré-existentes e mídias adicionadas dinamicamente ao documento NCL utilizando comandos de edição *NCLEdit* (ABNT, 2008b).

O comandos de *NCLEdit* permitem que scripts NCLua alterem a apresentação de um documento NCL em tempo de execução, alterando suas configurações de sincronismo entre mídias sem que seja preciso gerar um novo documento NCL. No caso do *GingaStore*, as novas mídias são na verdade outras aplicações NCL executadas sobre a demanda do usuário. Através comandos *NCLEdit* o *GingaStore* inclui o documento NCL da aplicação desejada pelo usuário como um novo nó de contexto aninhado ao documento NCL do *GingaStore*.

O projeto de telas e a transição entre elas são descritas pelo diagrama de estados da Figura 23. A aplicação possui o seguinte conjunto de telas: “Menu Principal”, que apresenta as ações de recuperar e executar aplicações; “Adquirir Aplicações”, que lista aplicações compatíveis disponíveis no repositório; “Descrição de aplicação”, que exibe detalhes acerca de uma aplicação; “Instalando Aplicação”, que indica a recuperação e persistência da aplicação; e “Minhas aplicações”, que lista as aplicações instaladas e apresenta o estado da conectividade com os dispositivos necessários.

**Figura 23: Diagrama de estados do *GingaSpace***

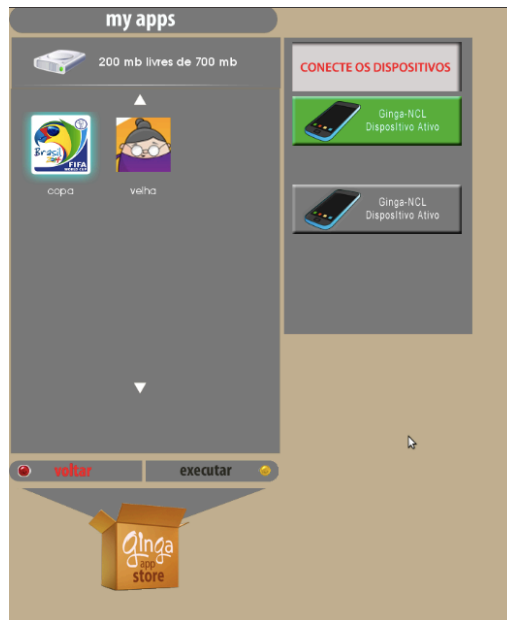


A apresentação de status da conectividade de dispositivos de segunda tela pelo *GingaStore* considera que as aplicações recuperadas do *GingaSpace* são compatíveis com o ambiente Ginga-NCL. Entretanto, os dispositivos de segunda tela requisitados por uma aplicação possuem um estado dinâmico de conectividade com o dispositivo receptor de TV Digital. Logo, o *GingaStore* a ser responsável por exibir uma representação gráfica da conectividade dos dispositivos de segunda tela.

Essa representação gráfica da conectividade dos dispositivos de segunda tela remete ao conceito apresentado pelo trabalho de Van Der Vlist et al. (2010). Esse trabalho propõe que dispositivos integrados de maneira ubíqua em uma residência necessitam apresentar ao usuário uma representação visual de conectividade entre eles.

A na Figura 24 ilustra a tela de “Minhas Aplicações” do protótipo desenvolvido. Ela apresenta informações que permitem a configuração dos dispositivos pelo usuário. Nela, para a execução da aplicação “Copa” são necessário dois dispositivos do tipo *ActiveClass*. No entanto, apenas um está conectado, o que impossibilita sua execução até que o segundo dispositivo esteja conectado.

Figura 24: Semântica da conectividade de dispositivos no GingaStore



## 5. Implementação e testes

Este capítulo tem o intuito complementar a descrição da arquitetura proposto o esquema “4+1” descrevendo a visão de implementação. Ela apresenta os casos de testes e o relato da execução dos testes do protótipo desenvolvido. Esse relatório permite validar os casos de uso, nos quais os testes foram baseados e conseqüentemente requisitos elencados para o sistema.

### 5.1. Visão de implementação

Essa visão tem o intuito de apresentar os elementos da arquitetura relacionados a configurações de *software* do sistema. Ela utiliza diagramas UML de pacotes para apresentar abstrações de classes e bibliotecas *software*.

O diagrama de pacotes apresentado na Figura 25 revela detalhes da implementação da aplicação *GingaStore*. O diagrama apresenta a estrutura e o relacionamento entre os pacotes de *software* para do sistema. A principal dependência da aplicação consiste no ambiente de execução Ginga-NCL com perfil EDTV. Esse ambiente deve possuir os seguintes módulos lua descritos na (ABNT, 2008b): o módulo *canvas* que permite criação das telas do *GingaStore*; o módulo *lua.io* que permite a persistência das aplicações recuperadas; e módulo *event* permite o uso de conexões *TCP* e realizar chamadas *NCLEdit* ao ambiente Ginga-NCL, viabilizando a reconfiguração dinâmica do *GingaStore*.

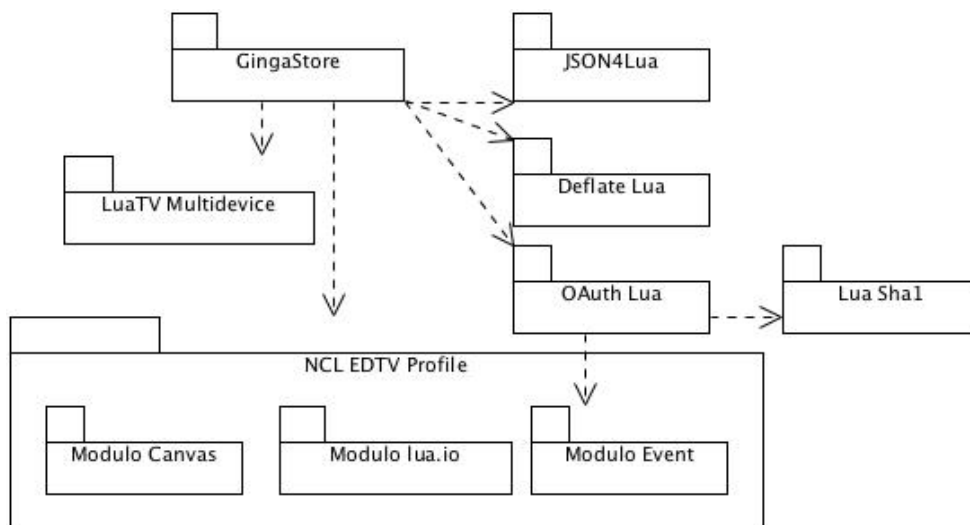
As outras dependências são três bibliotecas baseadas apenas em Lua, que são utilizadas para a recuperação das aplicações. Uma versão modificada da OAuth Lua<sup>1</sup> é usada para acesso à interface REST do repositório. O módulo de software recebeu modificações para ser compatível com a implementação de referência do *middleware* Ginga. A biblioteca JSON4Lua permite o tratamento de mensagens no formato JSON, que no sistema são providas das requisições REST

---

<sup>1</sup> <http://github.com/ignacio/LuaOAuth>. Acessado em: agosto de 2012.

para o repositório. E a biblioteca *Deflate Lua*<sup>1</sup> que permite a descompressão das aplicações armazenadas no repositório sobre o formato Deflate<sup>2</sup>.

**Figura 25: Diagrama de pacotes do *GingaStore***



Com o intuito de manter a compatibilidade com o Ginga foi necessário utilizar a biblioteca *Deflate Lua* e realizar modificações na implementação na biblioteca *OAuth Lua*. A escolha pela biblioteca *Deflate Lua* se deveu pela escolha compressão das aplicações no repositório com o formato GNU Zip<sup>3</sup> ao invés de Zip<sup>4</sup>. Enquanto a versão original da *OAuth Lua* utilizava bibliotecas *LuaSocket*<sup>5</sup> e *LuaCrypto*<sup>6</sup> não compatíveis com o Ginga. Elas foram substituídas respectivamente pelo módulo *NCLua TCP* do Ginga-NCL e pela biblioteca *Lua Sha1*<sup>7</sup>.

O módulo de multidispositivo da biblioteca *LuaTV* (BRANDAO, 2010) foi proposta no modelo de extensão de classes de dispositivos do Ginga-NCL (BATISTA et al., 2010). Ela permite a dinâmica adição e recuperação de quantidade dispositivos de classes definidas a partir de descrições RDF arbitrários dos desenvolvedores.

A biblioteca *LuaTV* não é normatizada como parte do ambiente de execução Ginga-NCL na especificação ISDB-Tb. Logo, a ausência dessa biblioteca (i.e. falha em chamadas *require lua*)

<sup>1</sup> <http://github.com/davidm/luas-compress-deflatelua>. Acessado em: agosto de 2012.

<sup>2</sup> <http://tools.ietf.org/html/rfc1951>. Acessado em: agosto de 2012.

<sup>3</sup> <http://www.gzip.org>. Acessado em: agosto de 2012.

<sup>4</sup> <http://www.ietf.org/rfc/rfc1950.txt>. Acessado em: agosto de 2012.

<sup>5</sup> <http://w3.impa.br/~diego/software/luasocket/>. Acessado em: agosto de 2012.

<sup>6</sup> <http://luacrypto.luaforge.net>. Acessado em: agosto de 2012.

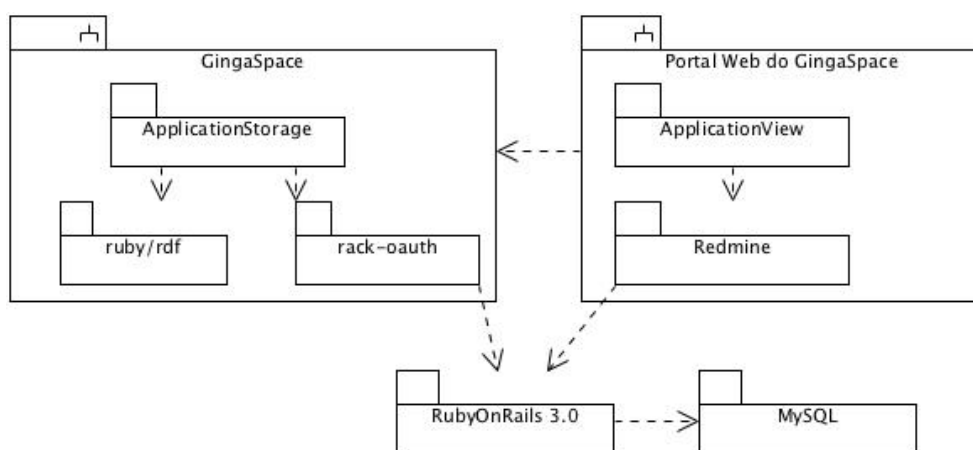
<sup>7</sup> <http://regex.info/blog/luas/sha1>. Acessado em: agosto de 2012.

em um contexto de execução implicaria em considerá-lo como ambiente padrão Ginga-NCL do ISDB-Tb. Ambiente padrão caracterizado por um perfil NCL EDTV, mídias padrões da especificação de dados e conectividades de dispositivos *gncl:deviceServices* do tipo *ActiveClass* e *PassiveClass*.

O diagrama de pacotes apresentado na Figura 26 revela detalhes da implementação do repositório GingaSpace, definindo a estrutura e relacionamento entre os pacotes de *software* do subsistema. Como descrito na visão de implantação, esse subsistema é composto de componentes de *software* GingaSpace e a o Portal *Web*. Eles são responsáveis respectivamente pelo serviço de armazenamento de aplicações e pela gestão das aplicações e desenvolvedores. Por serem ambos os serviços *Web*, eles são desenvolvidos através do framework *Web RubyOnRails*<sup>1</sup>, utilizam uma base dados MySQL e são executados sobre o servidor *Web* Apache, utilizando o modulo *Phusion Passenger*<sup>2</sup>.

Dentro da repositório GingaSpace a instância do framework *RubyOnRails* é realizada pelo uso do framework *RackOauth*<sup>3</sup>, que oferece uma gestão dos serviços REST com autenticação *OAuth* para o repositório. Nele o pacote *ApplicationStorage* modela os meta-dados e utiliza funcionalidades presentes no framework *RackOauth* para criar a interface REST de gerenciamento das aplicações.

**Figura 26: Diagrama de pacote do GingaSpace**



<sup>1</sup> <http://www.rubyonrails.org>. Acessado em: agosto de 2012.

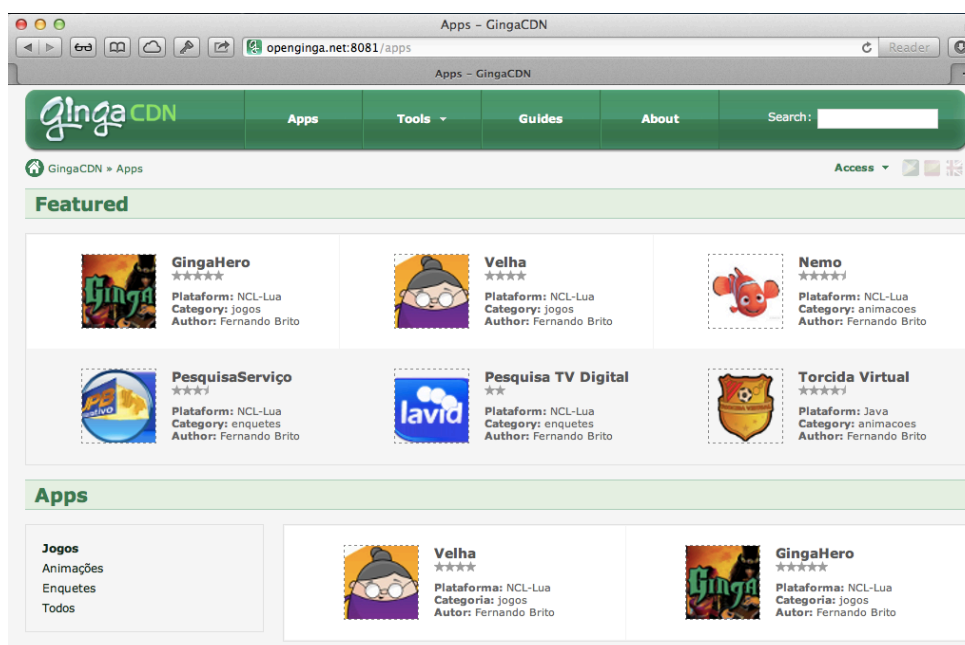
<sup>2</sup> <http://www.modrails.com>. Acessado em: agosto de 2012.

<sup>3</sup> <http://www.github.com/remi/rack-oauth>. Acessado em: agosto de 2012.

Dentro do Portal *Web* a instancia do *framework RubyOnRails* é realizada pelo uso do *framework* de gerência de projetos *Redmine*<sup>1</sup>, que permite funcionalidades de gestão de desenvolvedores, *issue track*, *wiki*, nacionalização, entre outros. O portal possui ainda o pacote *ApplicationView* utiliza funcionalidades presentes no *framework Redmine* para cria interfaces de controle e cadastro de aplicações.

Salienta-se que a decisão de utilizar o *Redmine* como ambiente de gestão dos desenvolvedores foi influenciada pelo projeto *GingaCDN*<sup>2</sup>. Logo o Portal Web do *GingaSpace* utiliza a base desenvolvedores existente e estende o portal *GingaCDN*. A Figura 27 ilustra o protótipo do Portal *Web* do Repositório visualizada em um navegador *Web*.

**Figura 27: Protótipo do Portal Web *GingaSpace***



O *GingaStore* foi desenvolvido utilizando uma versão da implementação de referência *Ginga-NCL*<sup>3</sup> com as funcionalidades de *NCLEdit* e o modulo IO de lua habilitadas. Entretanto, foram testes realizados na atualmente TV de LED LE5552I<sup>4</sup> da TOSHIBA que consiste na única

<sup>1</sup> <http://www.redmine.org>. Acessado em: agosto de 2012.

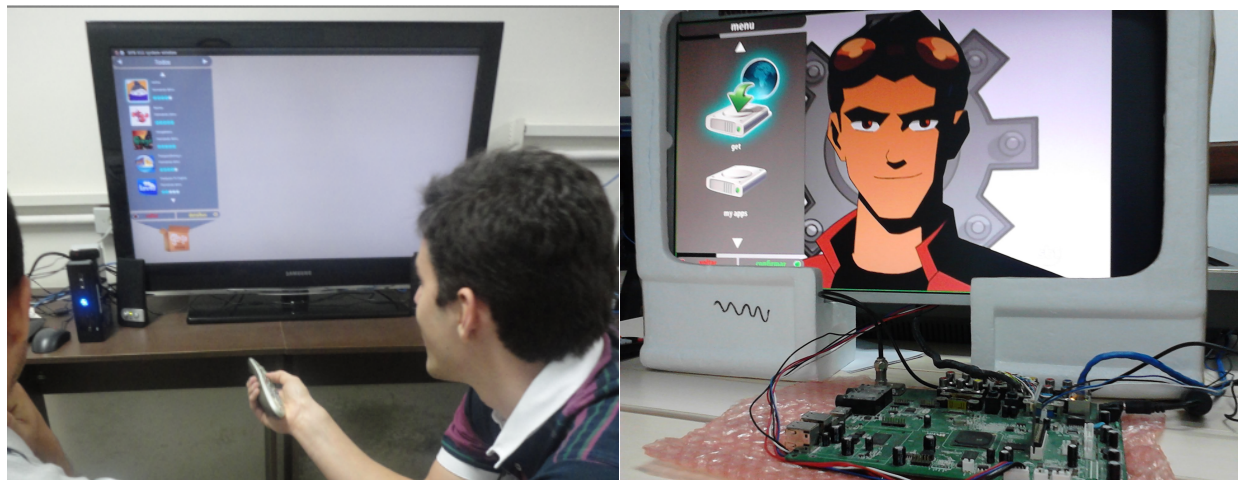
<sup>2</sup> <http://openginga.net>. Acessado em: agosto de 2012.

<sup>3</sup> [http://www.softwarepublico.gov.br/ver-comunidade?community\\_id=1101545](http://www.softwarepublico.gov.br/ver-comunidade?community_id=1101545). Acessado em: agosto de 2012.

<sup>4</sup> [http://www.semptoshiba.com.br/media/458235/05\\_le4652i\\_481360\\_p.pdf](http://www.semptoshiba.com.br/media/458235/05_le4652i_481360_p.pdf). Acessado em: agosto 2012

TV comercial com *middleware* Ginga embarcado com essas funcionalidades citadas. A Figura 28 ilustra a execução do protótipo nesses dois tipos de receptores.

**Figura 28 – Execução do protótipo do *GingaStore***



## 5.2. Casos de testes do sistema

A partir da visão de casos de uso, foram derivados os casos de testes listados nas 6, 7 e 8. Eles têm o intuito de validar a arquitetura proposta ao serem aplicados nos protótipos de *software* desenvolvidos.

A Tabela 6 lista casos de testes para cadastro de aplicações no repositório GingaSpace. Utilizando as aplicações ISDB-Tb APP1, APP2 com uso de 2 dispositivos *AciveClass*, APP3 com uso de mídia OGG e APP4 com uso de 1 dispositivo *UPnP*.

**Tabela 6: Casos de testes de cadastro de aplicações no *GingaSpace***

Ref.#	Caso de teste	
CT1	<b>Desenvolvedor realiza Cadastro de aplicação NCL com uso de mídias padrões do ISDB-Tb</b>	
	Pré-condições	<ul style="list-style-type: none"> <li>Desenvolvedor cadastrado no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>Desenvolvedor de posse de arquivo comprimido para aplicação APP1</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>Desenvolvedor acessa tela de cadastro no Portal <i>Web</i> do GingaSpace</li> </ol>

		<ol style="list-style-type: none"> <li>2. Desenvolvedor preenche formulário de cadastro de aplicação(nome de documento NCL principal, descrição, categoria, ícone representativo, link de vídeo)</li> <li>3. Desenvolvedor realiza <i>upload</i> de arquivo comprimido</li> <li>4. Desenvolvedor requisita cadastro</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>• Meta-dados e arquivo comprimida de APP1 armazenados no GingaSpace</li> </ul>
<b>CT1</b>	<b>Desenvolvedor realiza Cadastro de aplicação NCL com uso de mídias e classe de dispositivos <i>ActiveClass</i> padrões do ISDB-Tb</b>	
	Pré-condições	<ul style="list-style-type: none"> <li>• Desenvolvedor cadastrado no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>• Desenvolvedor de posse de arquivo comprimido de APP2</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>1. Desenvolvedor acessa tela de cadastro no Portal <i>Web</i> do GingaSpace</li> <li>2. Desenvolvedor preenche formulário de cadastro de aplicação (nome de documento NCL principal, descrição, categoria, ícone representativo, link de vídeo)</li> <li>3. Desenvolvedor realiza <i>upload</i> de arquivo comprimido</li> <li>4. Desenvolvedor requisita cadastro</li> <li>5. Portal <i>Web</i> percorre documento NCL principal, identifica uso apenas de mídias padrões do ISDB-Tb</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>• Meta-dados e arquivo comprimido de APP2 armazenados no GingaSpace</li> </ul>
<b>CT3</b>	<b>Desenvolvedor realiza Cadastro de aplicação NCL com uso de mídia <i>OGG</i></b>	
	Pré-condições	Desenvolvedor cadastrado no repositório.
	Entradas	Desenvolvedor de posse de aplicação APP3 comprimida
	Procedimentos	<ol style="list-style-type: none"> <li>1. Desenvolvedor acessa tela de cadastro no Portal <i>Web</i> do GingaSpace</li> <li>2. Desenvolvedor preenche formulário de cadastro de aplicação (nome de documento NCL principal, descrição, categoria, ícone representativo, link de vídeo)</li> <li>3. Desenvolvedor realiza upload de arquivo comprimida</li> <li>4. Desenvolvedor requisita cadastro de APP3</li> <li>5. Portal <i>Web</i> percorre documento NCL principal, identifica a uso de mídia não padrão do ISDB-Tb e cria RDF de descrição de receptor de TV com <i>prf:Ccpp-accept</i> com <i>OGG</i>.</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>• Meta-dados e arquivo comprimida de APP3 armazenados no GingaSpace</li> </ul>
<b>CT4</b>	<b>Desenvolvedor realiza Cadastro de aplicação NCL com uso de classe de dispositivo <i>UPnP</i></b>	
	Pré-condições	<ul style="list-style-type: none"> <li>• Desenvolvedor cadastrado no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>• Desenvolvedor de posse da aplicação APP4 comprimida</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>1. Desenvolvedor acessa tela de cadastro no Portal <i>Web</i> do GingaSpace</li> <li>2. Desenvolvedor preenche formulário de cadastro de aplicação(nome de documento NCL principal, descrição,</li> </ol>

		categoria, ícone representativo, link de vídeo) 3. Desenvolvedor realiza upload de arquivo comprimida 4. Desenvolvedor requisita cadastro de APP4 5. Portal <i>Web</i> percorre documento NCL principal, identifica a uso de dispositivo <i>UPnP</i> e cria RDF de descrição de receptor de TV com <i>gncl:deviceService</i> com <i>UPnP</i>
	Resultados esperados	<ul style="list-style-type: none"> <li>• Meta-dados e arquivo comprimida de APP4 armazenados no GingaSpace</li> </ul>

A Tabela 7 descreve casos de testes usados para verificar a recuperação de aplicações no repositório *GingaSpace* pela aplicação *GingaStore*.

**Tabela 7: Casos de testes de *GingaStore* recuperando de lista de aplicações compatíveis no *GingaSpace***

Ref. #	Caso de teste	
<b>CT5</b>	<b><i>GingaStore</i> em ambiente Ginga-NCL ISDB-Tb recupera do Repositório lista de aplicações compatíveis</b>	
	Pré-condições	<ul style="list-style-type: none"> <li>• <i>GingaStore</i> possui chave de acesso cadastrada no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>• O repositório contem as aplicações: APP1, APP2, APP3 e APP4</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>1. Usuário acessa tela “Adquirir aplicações”</li> <li>2. <i>GingaStore</i> realiza requisição com <i>RDF</i> de descrição padrão de ambiente ISDB-Tb</li> <li>3. <i>GingaStore</i> apresenta lista de aplicações</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>• <i>GingaStore</i> lista de aplicações composta por APP1 e APP2.</li> </ul>
<b>CT6</b>	<b><i>GingaStore</i> em ambiente Ginga-NCL com suporte a mídia extra recupera do lista de aplicações compatíveis</b>	
	Pré-condições	<ul style="list-style-type: none"> <li>• <i>GingaStore</i> possui chave de acesso cadastrada no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>• O repositório contem as aplicações: APP1, APP2, APP3 e APP4</li> <li>• <i>LuaTV</i> oferece <i>RDF</i> com <i>EDTV</i> em <i>gncl:supportedNCLProfiles</i> e <i>OGG</i> em <i>prf:CcppAccept</i>.</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>1. Usuário acessa tela “Adquirir aplicações”</li> <li>2. <i>GingaStore</i> realiza requisição com <i>RDF</i> de descrição do ambiente recuperado pelo <i>LuaTV</i>.</li> <li>3. <i>GingaStore</i> apresenta lista de aplicações</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>• <i>GingaStore</i> lista de aplicações composta por APP1, APP2 e APP3</li> </ul>
<b>CT7</b>	<b><i>GingaStore</i> em ambiente Ginga-NCL com suporte a mídia extra e a serviço <i>UPnP</i> recupera lista aplicações compatíveis</b>	
	Pré-condições	<ul style="list-style-type: none"> <li>• <i>GingaStore</i> possui chave de acesso cadastrada no repositório.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>• O repositório contem as aplicações: APP1, APP2, APP3 e APP4</li> </ul>

		<ul style="list-style-type: none"> <li><i>LuaTV</i> oferece <i>RDF</i> com <i>EDTV</i> em <i>gncl:supportedNCLProfiles</i>, <i>UPnP</i> em <i>gncl:supportedDeviceService</i> e <i>OGG</i> em <i>prf:CcppAccept</i>.</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>Usuário acessa tela “Adquirir aplicações”</li> <li><i>GingaStore</i> realiza requisição com <i>RDF</i> de descrição do ambiente <i>Ginga-NCL</i> pelo <i>LuaTV</i>.</li> <li><i>GingaStore</i> apresenta lista de aplicações</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li><i>GingaStore</i> lista de aplicações composta por : APP1, APP2, APP3 e APP4</li> </ul>

A Tabela 8 descreve casos de testes usados para verificar uso de listagem e execução de aplicações na aplicação *GingaStore*.

**Tabela 8: Casos de testes de da aplicação *GingaStore***

<b>Ref. #</b>	<b>Caso de teste</b>	
<b>CT8</b>	<b><i>GingaStore</i> instala aplicação recuperada do repositório</b>	
	Pré-condições	<ul style="list-style-type: none"> <li><i>GingaStore</i> possui chave de acesso cadastrada no repositório.</li> <li>Ambiente de execução <i>Ginga-NCL ISDB-Tb</i></li> </ul>
	Entradas	<ul style="list-style-type: none"> <li>O repositório contém a aplicação APP1 e APP2.</li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>Usuário acessa tela “Adquirir aplicações”</li> <li><i>GingaStore</i> realiza requisição com <i>RDF</i> de descrição padrão de ambiente <i>ISDB-Tb</i></li> <li><i>GingaStore</i> apresenta lista de aplicações com APP1 e APP2</li> <li>Usuário acessa tela detalhes da aplicação APP2</li> <li>Usuário requisita instalação da aplicação APP2</li> <li><i>GingaStore</i> apresenta tela de “Instalando Aplicação” e realiza a recuperação da APP1 no repositório.</li> <li><i>GingaStore</i> apresenta tela de “Adquirir Aplicações”</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>A aplicação APP1 é persistida e fica disponível na tela de “Minhas Aplicações”</li> </ul>
<b>CT9</b>	<b><i>GingaStore</i> em ambiente <i>ISDB-Tb</i> executa aplicação instalada</b>	
	Pré-condições	<ul style="list-style-type: none"> <li><i>GingaStore</i> em ambiente <i>ISDB-Tb</i></li> </ul>
	Entradas	<ul style="list-style-type: none"> <li><i>GingaStore</i> possui instalado a aplicação H.761 APP1 e APP2 com uso de 2 dispositivos <i>ActiveClass</i></li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>Usuário acessa tela “Minhas aplicações”</li> <li>Usuário seleciona aplicação APP1.</li> <li><i>GingaStore</i> não apresenta ícones de status de conectividade para de dispositivos</li> <li>Usuário requisita a execução da aplicação APP1</li> <li><i>GingaStore</i> apresenta tela de “Minhas Aplicações”</li> </ol>

	Resultados esperados	<ul style="list-style-type: none"> <li>A aplicação APP1 é executada e ao seu termino e <i>GingaStore</i> apresenta tela “Minhas Aplicações”</li> </ul>
<b>CT10</b>	<b><i>GingaStore</i> em ambiente ISDB-Tb executa aplicação com uso de segunda tela instalada</b>	
	Pré-condições	<ul style="list-style-type: none"> <li><i>GingaStore</i> em ambiente</li> <li>Usuário possui dois dispositivos do tipo <i>ActiveClass</i>.</li> </ul>
	Entradas	<ul style="list-style-type: none"> <li><i>GingaStore</i> possui instalado a aplicação H.761 APP1 e APP2 com uso de 2 dispositivos <i>ActiveClass</i></li> </ul>
	Procedimentos	<ol style="list-style-type: none"> <li>Usuário acessa tela “Minhas aplicações”</li> <li>Usuário seleciona aplicação APP2</li> <li><i>GingaStore</i> indica o requisitos de dois dispositivos <i>ActiveClass</i> para execução da APP2</li> <li>Usuário inicializa serviços de <i>ActiveClass</i> nos dispositivos</li> <li><i>GingaStore</i> identifica que dispositivos requisitados pela aplicação estão conectados [Procedimentos Excepcional 1]</li> <li><i>GingaStore</i> indica a conectividade dos dois dispositivos <i>ActiveClass</i> requisitados da APP2</li> <li>Usuário requisita a execução da aplicação.</li> </ol>
	[Procedimentos Excepcional 1]	<ol style="list-style-type: none"> <li><i>GingaStore</i> identifica que um ou os dois dispositivos requisitados não estão conectados</li> <li><i>GingaStore</i> indica ausência de conectividade um ou os dois requisitados</li> <li>É bloqueado a requisição de execução da aplicação</li> </ol>
	Resultados esperados	<ul style="list-style-type: none"> <li>A aplicação APP2 é executada e ao seu termino e <i>GingaStore</i> apresenta tela “Minhas Aplicações”</li> </ul>

### 5.3. Relatório de execução dos casos de testes

Os casos de testes apresentados foram aplicados sobre o protótipo de *software* desenvolvido. A aprovação dos testes implica na validação dos casos de uso e conseqüentemente dos requisitos do sistema. A Tabela 9 a seguir apresenta as configurações dos equipamentos utilizados para executar o sistema durante a execução dos casos de teste que viabiliza a replicabilidade dos testes

**Tabela 9: Configurações de equipamentos utilizado na execução dos testes sobre o sistema**

Ref #	Dispositivo	Configuração
Serv1	Servidor do GingaSpace	Intel(R) Xeon(R) CPU X3363 Dual-Core 2.83GHz, 3GB RAM, Gigabit Ethernet, Ubuntu <sup>1</sup> 10.04 X86

<sup>1</sup> <http://www.ubuntu.com>. Acessado em: agosto de 2012

Serv2	Servidor do portal <i>Web</i> do <i>GingaSpace</i>	Intel(R) Xeon(R) CPU X3363 Dual-Core 2.83GHz, 3GB RAM, Gigabit Ethernet, Ubuntu <sup>1</sup> 10.04 X86
Maq1	PC com Ambiente <i>Ginga-NCL</i>	MiniPC AOpen <sup>1</sup> . Code 2 Duo, 2.13 GHz, 3GB RAM, Gigabit Ethernet, Ubuntu <sup>1</sup> 10.04
TV1	TV para saída de vídeo para PC com ambiente <i>Ginga-NCL</i>	TV PLASMA 42’’ SAMSUNG, Resolução Full HD
Mov1	Dispositivo móvel com <i>Ginga-NCL ActiveClass I</i>	Motorola “DEFY, 3.7”, Android 2.3, Processador Dual-Core 1GHz; 512MB RAM com
Mov2	Dispositivo móvel com <i>Ginga-NCL ActiveClass II</i>	Motorola “Milestone 2” Android 2.3 Processador Dual-Core 1GHz; 512MB RAM com

A execução dos casos de testes CT1, CT2, CT3 e CT4, correspondem respectivamente ao cadastro de quatro aplicações (APP1, APP2, APP3 e APP4) no Portal *Web* do *GingaSpace*. A APP1 chamada de “Jogo da Velha”<sup>4</sup> é um aplicação ISDB-Tb sem uso de dispositivos de segunda tela. A APP2 chamada “Copa do Mundo” (GUIMARÃES, 2012) é uma aplicação ISDB-Tb e utiliza dois dispositivos de segunda tela com serviço *Ginga-NCL ActiveClass*.

A APP2 possui quatro interatividades que ocorrem nos dispositivos paralelamente a um vídeo H.264 na tela do receptor que mostra cenas de jogos do Brasil. A primeira interatividade permite aos usuários navegar por uma interface que sumariza os campeões das edições da Copa do Mundo de futebol. No segundo ponto, é simulada a compra de pacotes turísticos para a Copa de 2014.

A terceira ocorre sincronizada com um pênalti apresentado no vídeo principal. Nela, um usuário simula o goleiro e o chutador do pênalti. Em seguida, o resultado da simulação é apresentado no tela do receptor de TV. A última interatividade apresenta um vídeo demonstrativo das regiões do país, fazendo referência a sede da copa de 2014.

As aplicações APP3 e APP4 são versões adicionais para a aplicação “Copa”. A APP3 é a versão que tem o vídeo principal em formato OGG ao invés de H.264, enquanto que a APP4 altera a quarta interatividade da aplicação utilizando uma classe de dispositivo que requer um serviço *UPnP* ao invés de usar a classe ativa da implementação de referência (serviço *ActiveClass*).

<sup>1</sup> <http://aopen.com>. Acessado em: agosto de 2012

<sup>2</sup> <http://gingamobile.lprm.inf.ufes.br>. Acessado em: agosto de 2012

<sup>3</sup> <http://gingamobile.lprm.inf.ufes.br>. Acessado em: agosto de 2012

<sup>4</sup> <http://clubencl.org.br/node/75>. Acessado em: agosto de 2012

Dado o atual *status* de desenvolvimento da *API LuaTV*, a execução dos casos de testes CT5, CT6 e CT7 utilizaram arquivos RDF recuperados estaticamente pela API. Enquanto a execução dos casos de testes CT8, CT9 e CT10 foi realizada em parceria com testes associados ao trabalho de Guimarães (2012). Esse trabalho tem o intuito de avaliar a experiência de usuário e a usabilidade das aplicações de segunda tela e o uso dessas no *GingaStore*.

Os testes foram aplicados a um conjunto de 22 usuários organizados em 11 duplas. A Figura 29 ilustra o local onde foram aplicados os testes, o qual foi montado com intuito de se aproximar de um local destinado ao consumo de conteúdo de TV (uma sala de estar residencial). Cada dupla possuía dispositivos secundários com serviços *Ginga-NCL AtiveClass*, um controle remoto e um receptor *Ginga*, compatível com ISDB-Tb e executando o *GingaSpace*.

**Figura 29: Configuração de equipamentos para execução dos testes**



Dentre os questionamentos de avaliação de experiências de usuário realizados por Guimarães (2012), um é considerado importante para esta dissertação, o qual está associado à facilidade de entendimento do uso de dispositivos de segunda tela a partir do *GingaStore*. Os resultados indicam que aproximadamente 77% dos usuários acharam fácil entender quando é necessário conectar algum dispositivo para executar a aplicação. Desse modo, é possível concluir que o *GingaStore* conseguiu utilizar abstrações corretas para representar a conectividade com dispositivos de segunda tela.

## 6. Considerações Finais

Este capítulo sumariza o trabalho desenvolvido na presente dissertação de mestrado e extrai algumas considerações dos resultados obtidos. Esta seção também enfatiza a contribuição realizada dentro do ecossistema de TV Digital interativa e face os trabalhos relacionados, finalizando com a apresentação de perspectivas de evolução para a solução proposta.

### 6.1. Trabalho desenvolvido

Esta dissertação alcança os objetivos propostos no capítulo inicial a medida que: descreve uma a arquitetura de *Application Store* que considera a heterogeneidade multidispositivos do ambiente de execução de cada usuário (receptores de TV e dispositivos de segunda tela); descreve e implementa protótipos de software dessa arquitetura; e realiza execução de casos de teste sobre os protótipos com intuito de validar a arquitetura proposta.

Dois subsistemas compõem essa arquitetura de *Application Store*. O subsistema *GingaSpace* é responsável por armazenar aplicações NCL, mantendo uma descrição *UAProf* do ambiente de execução Ginga-NCL, sem restrição de mídias e classes de dispositivo requisitadas. E a *GingaStore* consiste em uma aplicação Ginga-NCL dinamicamente reconfigurável (DE SOUSA JUNIOR, 2012), que executa outras aplicações NCL recuperadas do *GingaSpace*, compatíveis com seu contexto de execução Ginga-NCL.

Esta dissertação utiliza o ambiente Ginga-NCL como meio de interoperabilidade multidispositivos, suportando diferentes exibidores de mídia e classes de dispositivos de segunda tela. Enquanto que os trabalhos relacionados com *Application Stores* utilizam estratégias de aplicações de segunda tela limitadas a configurações proprietárias ou muito específicas de receptor e dispositivos.

A cerca da descrição de dispositivos, esta dissertação não oferece o mesmo grau de complexidade semântica encontrado em outros trabalhos para a descrição de dispositivos. Entretanto ela oferece um descrição não restritiva e compatível com a utilizada pela

implementação de referência do *middleware* Ginga. A Tabela 10 resume as diferenças da abordagem proposta pelo *GingaSpace* e esses trabalhos relacionados.

**Tabela 10: Comparação com trabalhos relacionados considerando o *GingaSpace***

<b>Trabalho</b>	<b>Formato de aplicações do receptor</b>	<b>Formato de aplicações de segunda tela</b>	<b>Descrição de requisitos de dispositivo</b>	<b>Descoberta de dispositivos</b>
<i>Yahoo Conencted TV</i>	<i>JavaScript</i>	Limitadas ao protocolo <i>Knox</i>	Estático	<i>mDNSResponde</i>
<i>Google TV</i>	<i>Android</i>	Limitadas ao protocolo <i>Anymote</i>	Estático	<i>Anymote</i>
<i>Sticker Center</i>	<i>Lua</i>	Aplicação <i>Web</i>	Estático	Proprietário
(TOGIAS et al., 2010)	-	-	<i>Physical Device Ontology</i>	<i>UPnP</i>
(LI et al., 2008)	-	-	<i>MPEG-21 DIA</i>	Dependente de transformação <i>XSLT</i>
<i>GingaSpace</i>	Ginga-NCL	Não restritivo	<i>UAProf</i> com extensões de (BATISTA et al., 2010)	Não restritivo

## 6.2. Contribuições

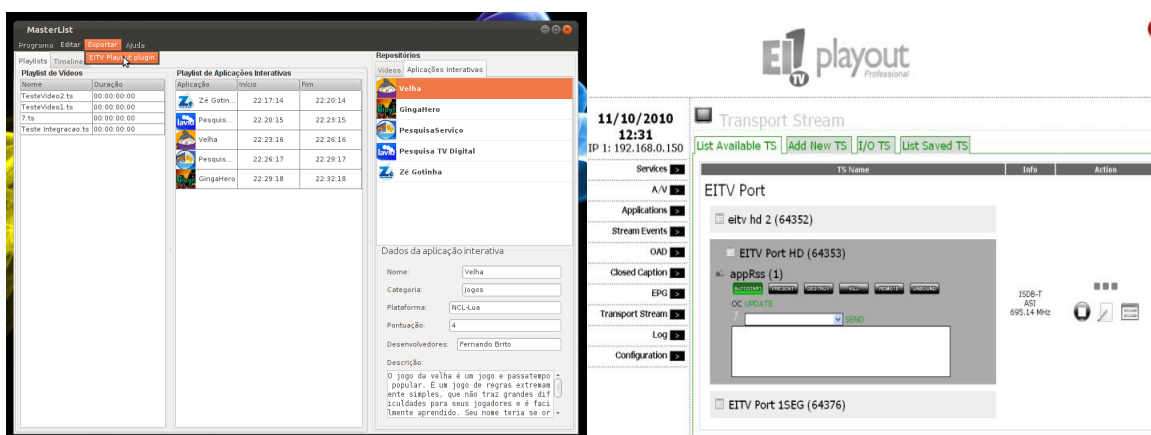
A principal contribuição desta dissertação foi criar um novo cenário de uso para a TV Digital interativa, que integra os cenários de *Broadband TV* e aplicações de segunda tela. Construindo validações aos trabalhos de Batista et al. (2010) e de Sousa Junior (2012), ao realizar essa integração através do ambiente Ginga-NCL.

Nesse novo cenário o repositório *GingaSpace* permite assumir responsabilidade de distribuir as aplicações compatíveis para cada usuário, na medida em que os desenvolvedores associem requisitos multidispositivo a suas aplicações. Removendo a necessidade de adaptação ou recriação de aplicações para os diferentes contextos de multidispositivos. E adicionalmente oferecendo através da aplicação *GingaStore* que o usuário de TV execute apenas aplicações compatíveis ao seu ambiente de execução e com suporte visual da conectividade dos dispositivos de segunda tela requisitados.

Contribuições colaterais foram realizadas em esforços relacionadas ao projeto *GingaAppStore* (DE SOUZA FILHO, 2009). Primeiro, ao prover um cenário de uso do

*GingaSpace* dentro de emissoras de TV Digital ao utilizar a ferramenta chamada de *GingaAppStore Playlist*. Essa ferramenta utiliza um *Playout*(EITV) para criar um canal de radio difusão conteúdo conteúdo audiovisual e aplicações advindas do *GingaSpace*, como é ilustrado na Figura 30.

**Figura 30: *GingaAppStore Playlist* e configuração de EITV *Playout***



### 6.3. Trabalhos futuros

Os primeiros trabalhos a serem considerados a partir desta dissertação são: desenvolver a funcionalidade de cadastro de aplicação como um plug-in da ferramenta de autoria *NCL Composer*<sup>1</sup>; desenvolver uma versão do *GingaStore* para executar em implementações para dispositivos portáteis do ambiente *Ginga-NCL*, como o *GingaMobile*<sup>2</sup>; e incrementar as descrições *UAProf* com descrições *OWL*. Esse incremento tem o intuito de abordar a problemática apontada por Togias et al. (2010), da falta de semântica para realizar raciocínios com base nas atuais descrições de dispositivos. As descrições *OWL* viabilizariam realizar inferências sobre compatibilidade de recursos multidispositivos atuais e futuras dos dispositivos.

Outros possíveis trabalhos podem avaliar modelos de negócios para cenários de uso da arquitetura proposta. Por exemplo, o cenário de uso que chamamos “*Application Store Channel*”, que considera a característica de portabilidade da aplicação *GingaStore* para realizar o envio do desta por *broadcast* como um conteúdo interativo *Ginga*. Esse cenário permite avaliar modelos de negócio como o proposto por Ramos (2012), para rentabilização por parte dos

<sup>1</sup> <http://composer.telemidia.puc-rio.br/>. Acessado em: agosto de 2012

<sup>2</sup> <http://gingamobile.lprm.inf.ufes.br>. Acessado em: agosto de 2012

*broadcast* em distribuir aplicações de patrocinadores. Esse modelo é semelhante ao modelo *Google Adwords*<sup>1</sup>, que possui links de patrocinadores em conteúdos distribuído pelo *Google*. Um esforço realizado nesse sentido foram testes do *GingaStore* no ambiente da TOSHIBA ilustrado anteriormente.

---

<sup>1</sup> <http://adwords.google.com>. Acessado em: agosto de 2012

## Referências

ABNT. 2008a. ABNT NBR 15606-1:Televisão Digital Terrestre — Codificação De Dados E Especificações De Transmissão Para Radiodifusão Digital Parte 1: Codificação De Dados. Disponível em: <<http://www.abntcolegao.com.br>>. Acessado em: agosto de 2012.

ABNT, 2008b. ABNT NBR 15606-2: Televisão Digital Terrestre – Codificação De Dados E Especificações De Transmissão Para Radiodifusão Digital Parte 2: Ginga-NCL Para Receptores Fixos E Móveis - Linguagem De Aplicação XML Para Codificação De Aplicações. Disponível em: <<http://www.abntcolegao.com.br>>. Acessado em: agosto de 2012.

ARIB. 2004. ARIB STD-B23 Version 1.2: Application Execution Engine Platform for Digital Broadcasting. Disponível em: <[http://www.arib.or.jp/english/html/overview/archives/br/ARIB\\_STD-B23\\_v1.1\\_E1.pdf](http://www.arib.or.jp/english/html/overview/archives/br/ARIB_STD-B23_v1.1_E1.pdf)>. Acessado em: agosto de 2012.

BATISTA, C.E.C.F., SOARES, L.F.G., SOUZA FILHO, G.L., 2010. Estendendo o uso das classes de dispositivos Ginga-NCL. In Proceedings of WebMedia '10: 16th Brazilian Symposium on Multimedia and the Web.

BATISTA, C.E.C.F., ANJOS, T.C., OMAIA, D., ARAÚJO T.M.U., BRASILEIRO F.V., SOUZA FILHO, G.L., 2008. TVGrid: Computação Em Grid Utilizando Recursos de Uma Rede De TV Digital. In Proceedings of WebMedia '08: 14th Brazilian Symposium on Multimedia and the Web.

BECKER, V., 2006. Concepção e Desenvolvimento de Aplicações Interativas para Televisão Digital. Dissertação(Mestrado), Universidade Federal de Santa Catarina.

BERNERS-LEE, T. H., 2001. The Semantic Web. Scientific American, Maio de 2001, p. 29-37.  
BOQUIMPANI, A., 2012. O Ginga Integrado com Novas Tecnologias Cloud, Multiscreen, Smart Devices. In Proceedings of TDC'2012 Developer Conference. São Paulo.

BRANDAO R.R.M., SOUZA FILHO, G.L., BATISTA, C.E.C.F., 2010. Extended Features for the Ginga-NCL Environment Introducing the LuaTV API. In Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN).

CHORIANOPOULOS, K. 2004. Virtual Television Channels: Conceptual Model, User Interface Design and Affective Usability Evaluation. e University of Economics and Business.

COSTA, R.M.R., MORENO, M.F., SOARES, L.F.G., 2009. Ginga-NCL: Suporte a Múltiplos Dispositivos. In Proceedings of WebMedia '09: 15th Brazilian Symposium on Multimedia and the Web.

CRUICKSHANK, L., TSEKLEVES, E., WHITHAM, R., Annette Hill, and Kaoruko Kondo. 2007. Making Interactive TV Easier to Use: Interface Design for a Second Screen Approach. The Design Journal, Vol. 10, No. 3.

SILVA, M.A.M., CASTILHOS, E.L., EICHLER, F.A.V., NOGUEIRA, R.F. 2010. T-Government: O Projeto TV Digital – Social. Um case de oferta de serviços Interativos para os Cidadãos. In Proceedings of WebMedia '10: 16th Brazilian Symposium on Multimedia and the Web.

DE SOUSA JUNIOR, J. G, SOARES, L.F.G., 2012. Uma Arquitetura para Aplicações Dinâmicas NCL Baseadas em Famílias de Documentos. Dissertação(Mestrado), Pontifícia Universidade Católica do Rio de Janeiro.

DE SOUZA FILHO, G.L., LEITE, L.E.C., BATISTA, C.E.C.F, 2007. Ginga-J: The Procedural *Middleware* for the Brazilian Digital TV System. Journal of Brazilian Computer Society, Vol.12, No. 4.

DE SOUZA FILHO, G.L., 2009. GingaAppStore: Um framework para desenvolvimento e disponibilização de aplicações interativas para SBTVD - Anexo I: Proposta Técnica Detalhada.

FERN, A., REBECA, P.D., CABRER, M.R., ARIAS, J.P., SOLLA, A.G., GARC, J., FERN, Y.B., 2006. MHP-OSGi Convergence : a New Model for Open Residential Gateways. Software: Practice and Experience, Vol. 36, No. 13, p 1421–1442.

FERNANDES, J., SILVEIRA, G., SOUZA FILHO, G.L., 2004. Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas. Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, JAI-SBC.

FRKOVIC, F., PODOBNIK V., TRZEC, K., JEZIC, G., 2008. “Agent-Based User Personalization Using Context- Aware Semantic Reasoning”. In Proceedings of KES '08: 12th international conference on Knowledge-Based Intelligent Information and Engineering System.

GAWLINSKI, M. 2003. Interactive television production. Focal Press.

GHISI, Bruno Cavaler. 2010. “Conceptual Models for T-Commerce in Brazil”. In Proceedings of EuroITV 2010: Workshop on Interactive Digital TV in Emergent Countries.

GHISI, Bruno Cavaler. 2010b. “Integração de Aplicações Para TV Digital Interativa com Redes Sociais”. In Proceedings of I WTVDI: Workshop de TV Digital Interativa.

GOOGLE. 2012. Guia De Desenvolvimento para GoogleTV. Disponível <<https://developers.google.com/tv/>>. Acessado em agosto de 2012.

GUIMARÃES, A.P.N. Experiência de Uso de Dispositivos Convergentes na TV Digital Brasileira: Um Estudo de caso baseado no Ginga, 2012. Dissertação(Mestrado), Universidade Federal da Paraíba.

ITU, 2001. ITU-T Rec. J.200: Worldwide common core – Application environment for digital interactive television services, Disponível em: <<http://www.itu.int/rec/T-REC-J.200>>. Acessado em: agosto de 2012.

ITU, 2001. ITU-T Rec. J.200: Worldwide common core – Application environment for digital interactive television services, Disponível em: <<http://www.itu.int/rec/T-REC-J.200>>. Acessado em: agosto de 2012.

ITU, 2009a. ITU-T Rec. H.760: IPTV multimedia services and applications for IPTV, Disponível em: <<http://www.itu.int/rec/T-REC-H.760>>. Acessado em: agosto de 2012.

ITU, 2009b. ITU-T Rec. H.761: Overview of Multimedia Application Frameworks for IPTV, Disponível em: <<http://www.itu.int/rec/T-REC-H.761>>. Acessado em: agosto de 2012.

ITU, 2009c. ITU-T Rec. Y.1910: IPTV functional architecture. Disponível em: <<http://www.itu.int/rec/T-REC-Y.1910/en>>. Acessado em: agosto de 2012.

ITU, 2011. ITU-T Rec. H.762: Lightweight interactive multimedia framework for IPTV services (LIME), Disponível em: <<http://www.itu.int/rec/T-REC-H.762>>. Acessado em: agosto de 2012.

ITU, 2012. What is IPTV?. Disponível em: <<http://academy.itu.int/index.php/topics/item/328-iptv>>. Acessado em: agosto de 2012.

KRUCHTEN, P.B. 1995. The 4+1 View Model of Architecture. IEEE Software Jornal, Vol. No.12.

KULESZA, R. et al., 2011. Uma Abordagem Dirigida por Modelos para Integração de Aplicações Interativas e Serviços *Web*: Estudo de caso na Plataforma de TV Digital. In Proceedings of WebMedia '11: 17th Brazilian Symposium on Multimedia and the *Web*.

KULESZA, Raoni, C.A.S. Santos, T.A. Tavares, M.M. Neto, and Guido Lemos de Souza Filho. 2010. Desenvolvimento de Aplicações Imperativas Para TV Digital No *Middleware* Ginga Com *Java*. In Proceedings of WebMedia '10: 16th Brazilian Symposium on Multimedia and the *Web*.

LI, N., ATTOU A., DE, S., MOESSNER, K., 2008. Device and Service Descriptions for Ontology-Based Ubiquitous Multimedia Services. In Proceedings of MoMM '08: 6th International Conference on Advances in Mobile Computing and Multimedia.

MORENO, M.F., BATISTA, C.E.C.F., SOARES, L.F.G., 2010. NCL and ITU-T's Standardization Effort on Multimedia Application Frameworks for IPTV.” I WTVDI: Workshop de TV Digital Interativa.

NHK-STRL. 2012. The Evolution of TV - a Brief History of TV Technology in Japan. Disponível em: <<http://www.nhk.or.jp/strl/aboutstrl/evolution-of-tv-en>>. Acessado em: agosto de 2012.

ISO/IEC, 2000. ISO/IEC N3752: Overview of the MPEG-7 Standard (version 4.0).

ISO/IEC, 2004. ISO/IEC 21000-7: Information Technology - Multimedia Framework (MPEG-21) — Part 7: Digital Item Adaptation.

OPEN MOBILE ALLIANCE. 2001. WAG *UAProf* . Wireless Application Protocol WAP-248-*UAPROF*-20011020-A. Disponível em: <<http://www.openmobilealliance.org/Technical/wapindex.aspx>>. Acessado em: agosto de 2012

OSGI ALLIANCE, 2012. *OSGi* Core Release 5. *OSGi*. Disponível em: <<http://www.OSGi.org/Download/Release5>>. Acessado em: agosto de 2012.

PAUTASSO, C., ZIMMERMANN, O., LEYMANN, F., 2008. “RESTful Web Services vs. ‘Big’ Web Services: Making the Right Architectural Decision.” WWW '08: 17th international conference on World Wide Web

QUICO, C., 2003. Are communication services the killer applications for Interactive TV? or I left my wife because I am in love with the TV set. 1st European Conference on Interactive Television: from Viewers to Actors

RAMOS, D.G., 2012. Metaheurísticas GRASP E ILS aplicadas ao Problema da Variabilidade do Tempo de Download em Ambientes de TVDI. Dissertação(Mestrado), Universidade Federal da Paraíba.

STALLINGS, W., 2007. Criptografia E Segurança De Redes - Princípios e Práticas. 4th ed. Prentice Hall.

SAWANT, S., 2010. Mobile Marketing Strategies for B2C Companies. Dissertação(Mestrado), - Massachusetts Institute of Technology.

SCHMIDT, B., ADAMS, N., WANT, R., 1994. "Applications., Context-Aware Computing." 1st International Workshop on Mobile Computing Systems and Applications. Santa Cruz.

SEGUNDO, R.M.C., SILVA, J.C.F., TAVARES, T.A., 2010. ATHUS: A Generic Framework for Game Development on Ginga Middleware. In Proceedings of SBGAMES 2010: Brazilian Symposium on Games and Digital Entertainment.

SILVA, F.P.R., 2010. Xtation: Um Ambiente Para Execução e Teste de Aplicações Interativas para O *Middleware* Ginga. Dissertação(Mestrado), Universidade Federal da Paraíba.

SOARES, L.F.G., 2010. Towards the convergence of digital TV systems. Journal of Internet Services and Applications. Journal of Internet Services and Applications, Vol. 1, No. 1.

SOARES, L.F.G., RODRIGUES, R.F., MORENO, M.F., 2007. Ginga-NCL: the declarative environment of the Brazilian digital TV System. Journal of the Brazilian Computer Society, Vol. 12, p. 37-46.

SUN MICROSYSTEMS. 2008. *Java* DTV API 1.0 Specification. Disponível em: <<http://java.sun.com/javame/technology/javatv>>. Acessado em: agosto de 2012.

SUN MICROSYSTEMS. 2008b. *Java* TV Specification 1.1 - JSR 927. Disponível em: <<http://jcp.org/en/jsr/detail?id=927>>. Acessado em: agosto de 2012.

TAKECHI, M, A Baba, R Sawai, K Matsumura, and CLIENTE Shishikui. 2011. "Advanced Interactive Broadcasting Service Platform by Using *Java* Data Broadcasting."

TOGIAS, K., GOUMOPOULOS, C., 2010. Ontology-Based Representation of *UPnP* Devices and Services for Dynamic Context-Aware Ubiquitous Computing Applications. In Proceedings of CTRQ '10 Third International Conference on Communication Theory, Reliability, and Quality of Service.

TOTVSm 2010. "Sticker Center: Workshop de Aplicações Interativas." Publicado em Congresso SET 2010. Disponível em <<https://Cliente.astrodevnet.com/AstroDevNet/>>. Acessado em agosto de 2012.

UPNP FÓRUM, 2012. *UPnP* Device Architecture 1.1. Disponível em: <<http://www.upnp.org/>>. Acessado em: agosto de 2012.

VAN DER VLIST, B.J.J., HU, J., NIEZEN, G., FEIJS, L.M.G. 2010. "Design Semantics of Connections in a *Smart* Home Environment". DeSForM2010: Design and Semantics of Form and Movement, p. 48-56.

VANDEKAR, E., 2007. A Service System Design Approach for ITV Banking. In Proceedings of ICEC '07: Ninth international conference on Electronic commerce.

YAHOO, 2011. A New Model for Device Communication From Yahoo! Connected TV. API from Yahoo! v1.5. Disponível em: <<http://developer.yahoo.com/connectedtv/devicecommunication/YWEDeviceCommunicationWhitePaper.pdf>>. Acessado em: agosto de 2012.