

# DISSERTAÇÃO DE MESTRADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIA DA COMPUTAÇÃO

**“Arquitetura de um Provedor de Serviços  
Interativos para Sistemas de Televisão  
Digital”**

**ALUNO:** Gabriel Massote Prado  
**ORIENTADOR:** Prof. Dr. Sergio Donizetti Zorzo

São Carlos  
Março/2011

CAIXA POSTAL 676  
FONE/FAX: (16) 3351-8233  
13565-905 - SÃO CARLOS - SP  
BRASIL

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Arquitetura de um Provedor de Serviços Interativos  
para Sistemas de Televisão Digital**

**GABRIEL MASSOTE PRADO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Televisão Digital Interativa.  
Orientador: Prof. Dr. Sergio Donizetti Zorzo.

São Carlos - SP  
Março/2012

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

P896ap Prado, Gabriel Massote.  
Arquitetura de um provedor de serviços interativos para sistemas de televisão digital / Gabriel Massote Prado. -- São Carlos : UFSCar, 2013.  
63 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2011.

1. Software - desenvolvimento. 2. Televisão digital. 3. Serviços da WEB. I. Título.

CDD: 004.21 (20ª)

# Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

## “Arquitetura de um Provedor de Serviços Interativos para Sistemas de Televisão Digital”

GABRIEL MASSOTE PRADO

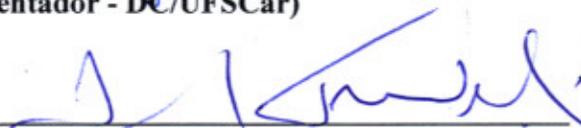
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:



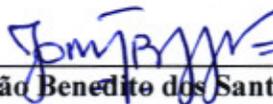
---

Prof. Dr. Sergio Donizetti Zorzo  
(Orientador - DC/UFSCar)



---

Prof. Dr. Luis Carlos Trevelin  
(DC/UFSCar)



---

Prof. Dr. João Benedito dos Santos Júnior  
(PUC/Poços de Caldas)

São Carlos  
Dezembro/2011

# AGRADECIMENTO

Agradeço primeiramente a Deus por me proporcionar todas as oportunidades que me foram concedidas até hoje; a minha família por estar sempre ao meu lado, apoiando-me quando necessário; especialmente, ao meu pai pelo total apoio durante a monografia de qualificação e na conclusão deste trabalho com esta dissertação; aos meus amigos que me ajudam, me divertem e me ensinam sempre coisas novas e boas na vida; em especial a Mariana Di Bella por tudo e por todo apoio durante o período de mestrado que passei em São Paulo; aos professores do Curso de Ciência da Computação da Universidade Federal de São Carlos pela base, oportunidades, confiança, amizade, qualidade do curso e por me ajudarem a chegar ao nível de conhecimento que possuo hoje; em especial, ao professor Sergio Donizetti Zorzo, que me ofereceu a oportunidade de trabalhar junto a ele durante todo o tempo que passamos juntos elaborando e desenvolvendo este trabalho sempre me passando confiança e me proporcionando toda a estrutura necessária; e especialmente também, junto ao professor Zorzo, ao meu caro chefe de trabalho, Michel Barros, pela compreensão e suporte a mim oferecidos todos os dias que precisava ir a São Carlos, possibilitando-me a conclusão do mestrado.

Agradeço a todos de uma maneira muito especial por tudo que fizeram por mim.

Que Deus abençoe todos!

# RESUMO

A evolução dos sistemas de televisão, de analógicos para digitais, proporciona uma nova e vasta gama de serviços diante da possibilidade de transmissão de dados junto à alta qualidade de áudio e vídeo. Com a agregação de dados a transmissão dos programas televisivos, surgem os sistemas de TV Digital Interativa permitindo a inclusão de conteúdos interativos implicando em um novo modo de produzir e transmitir os programas televisivos. Este trabalho apresenta uma Arquitetura de um Provedor de Serviços Interativos, nomeada *Interactive Service Provider for Digital Television Systems* – ispTV – para sistemas de televisão digital interativa, baseado em uma Arquitetura Orientada a Serviços que possibilita a publicação e oferecimento de serviços em diferentes tecnologias, tais como JAX-WS, Axis2 e serviços *ispTV*, com a inclusão de uma camada abstrata de comunicação com o intuito de simplificar o desenvolvimento de programas interativos junto a uma comunicação padronizada oferecida por APIs da arquitetura. O Sistema Brasileiro de Televisão Digital tem como um dos objetivos proporcionar a inclusão social, cultural e tecnológica através da televisão transmitindo aplicações educacionais, governamentais e comerciais para a interação do usuário com os programas televisivos. Neste contexto, este trabalho apresenta algumas aplicações interativas, educacionais e de suporte aos telespectadores com necessidades especiais, utilizando a arquitetura proposta. As aplicações apresentadas, evidenciam aspectos de usabilidade da arquitetura *ispTV* junto aos teste de usabilidade das APIs de abstração que a arquitetura oferece. Também são apresentados testes de desempenho e de escalabilidade da arquitetura, evidenciando que o seu funcionamento possibilita sua utilização para a transmissão de aplicações interativas para diversas classes de projetos, instituições e empresas.

**Palavras-chave:** Televisão Digital Interativa, Canal de Interatividade, Web service, Sistema Brasileiro de Televisão Digital, *Middleware*, Ginga.

# ABSTRACT

The evolution of television systems, analog to digital, provides a wide range of new services possibility of data transmission from the high quality audio and video. With the aggregation of data transmission of television programs, systems arise Interactive Digital TV allows you to embed interactive content resulting in a new way to produce and transmit television programs. This paper presents an architecture of an Interactive Service Provider named Interactive Service Provider for Digital Television Systems - ispTV - to interactive digital television systems, based on a service-oriented architecture that enables the publication and delivery of services in different technologies such as JAX-WS, Axis2 and ispTV services, with the inclusion of an abstract layer of communication in order to simplify the development of interactive programs with a standardized communication APIs provided by the architecture. The Brazilian Digital Television System is to provide an objective of social, cultural and technological inclusion through broadcasting educational, government and commercial television applications for the television user programs. In this context, this paper presents some interactive applications with educational support to viewers with special needs using the proposed architecture. The applications presented demonstrate the usability aspects of the architecture along ispTV to test usability of the APIs of abstraction that the architecture offers. Also presented are performance testing and scalability of the architecture, showing that its operation allows its use for the transmission of interactive applications for various classes of projects, institutions and companies.

**Keywords:** Brazilian System of Digital Television, Interactive Television, Interactive Channel, Ginga, Web service.

# LISTA DE FIGURAS

Figura 1: Sistema Interativo e Pseudo-Interativo.....	9
Figura 2: Arquitetura Set-Top Box.....	12
Figura 3: Arquitetura Middleware Ginga - adaptado de (GINGA, 2010) .....	13
Figura 4: Arquitetura de um Sistema de Televisão Digital Interativa com Canal de Interatividade adaptado de (ABNT NBR 15607-1, 2009) .....	15
Figura 5 - Requisitos de serviços e provedores de serviços - adaptado de (ERL, 2005) .....	18
Figura 6 - Papéis de Web services - adaptado de (ERL, 2005) .....	18
Figura 7 – Conceitos de interatividade referente a programas televisivos e aplicações interativas.....	20
Figura 8: Arquitetura do módulo do Provedor de Serviços Interativos .....	25
Figura 9: Protocolo com base TCP e UDP.....	30
Figura 10: Mensagem padrão ispTV .....	31
Figura 11: ispTV Web Tool.....	32
Figura 12: ispTV Web Tool.....	33
Figura 13: Fluxograma da arquitetura ispTV .....	36
Figura 14: Arquitetura Plataforma JiTV - adaptada de (SANTOS JUNIOR <i>et al.</i> , 2007) .....	38
Figura 15: Estrutura PSI com plataforma JiTV .....	39
Figura 16: Aplicação Interativa de Educação com menus para aplicações específicas .....	46
Figura 17: Fluxograma de interação.....	47
Figura 18: Aplicações Imersivas e Interativas utilizando <i>ispTV</i> .....	48
Figura 19: Aplicação Interativa XPTA.....	50
Figura 20: Utilização implementação Java API comunicação <i>ispTV</i> .....	55
Figura 21: Utilização implementação LUA API comunicação <i>ispTV</i> .....	56

# LISTA DE TABELAS

Tabela 1: Teste baseado apenas em delay (Tempo de Resposta) .....	60
Tabela 2: Teste baseado em processamento matemático .....	60
Tabela 3: Teste baseado em acesso a Banco de Dados .....	61
Tabela 4: Teste baseado nas requisições da arquitetura.....	61
Tabela 5: Cenários de teste .....	64

# LISTA DE ABREVIATURAS E SIGLAS

<b>SBTVD</b>	Sistema Brasileiro de Televisão Digital
<b>TVD</b>	Televisão Digital
<b>TVDI</b>	Televisão Digital Interativa
<b>STB</b>	<i>Set-Top Box</i>
<b>PSI</b>	Provedor de Serviços Interativos
<b>JiTV</b>	<i>Java Interactive Television</i>
<b>DVB</b>	<i>Digital Video Broadcasting</i>
<b>ATSC</b>	<i>Advanced Television Systems Committee</i>
<b>ISDB-T</b>	<i>Integrated Services Digital Broadcasting – Terrestrial</i>
<b>MHP</b>	<i>Multimedia Home Platform</i>
<b>DASE</b>	<i>DTV Application. Software Environment</i>
<b>ARIB</b>	<i>Association of Radio Industries and Businesses</i>
<b>GINGA-NCL</b>	<i>Ginga-Nexted Context Language</i>
<b>GINGA-JAVA</b>	<i>Ginga-Java</i>
<b>GINGA-CC</b>	<i>Ginga-Common Core</i>
<b>NCL</b>	<i>Nexted Context Language</i>
<b>ispTV</b>	<i>Interactive Service Provider for Digital Television Systems</i>
<b>ABNT</b>	Associação Brasileira de Normas Técnicas
<b>NBR</b>	Normas Brasileiras
<b>WSDL</b>	<i>Web Service Definition Language</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>JAX-WS</b>	<i>Java API for XML Web Services</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>EPG</b>	<i>Electronic Program Guides</i>
<b>TCP</b>	<i>Transmission Control Protocol</i>
<b>UDP</b>	<i>User Datagram Protocol</i>
<b>IP</b>	<i>Internet Protocol</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>TVDILAB</b>	Laboratório de Televisão Digital Interativa

<b>pTVDI</b>	Programas de Televisão Digital Interativa
<b>XML</b>	<i>eXtensible Markup Language</i>
<b>J2EE</b>	<i>Java 2 Platform, Enterprise Edition</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>REST</b>	<i>Representational State Transfer</i>
<b>UDDI</b>	<i>Universal Description, Discovery and Integration</i>
<b>WRCS</b>	<i>Web-based return channel</i>
<b>EJB</b>	<i>Enterprise JavaBeans</i>
<b>FAQ</b>	<i>Frequently Asked Questions</i>
<b>SQL</b>	<i>Structured Query Language</i>

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1 Considerações Iniciais.....	1
1.2 Motivação.....	3
1.3 Objetivos .....	4
1.4 Organização do Trabalho.....	5
<b>2. APLICAÇÕES INTERATIVAS EM TELEVISÃO DIGITAL INTERATIVA.....</b>	<b>7</b>
2.1 Aplicações Interativas .....	9
2.2 Set-Top Box .....	10
2.3 Canal de Interatividade.....	13
2.4 Web services.....	17
<b>3. ARQUITETURA DE UM PROVEDOR DE SERVIÇOS INTERATIVOS.....</b>	<b>22</b>
3.1 Arquitetura Genérica - <i>ispTV</i> .....	24
3.2 Mensagem Padrão .....	29
3.3 Web Tool.....	32
3.4 API Client .....	34
3.5 Trabalhos Relacionados.....	37
3.5.1 Canal de Retorno em Sistemas de Televisão Digital Interativa: Estratégias para a criação de aplicações utilizando um Provedor de Serviços Interativos .....	37
3.5.2 WRCS – Projeto de um Sistema de Canal de Retorno baseado na Web em broadcast de serviços de dados.....	40
<b>4. APLICAÇÕES INTERATIVAS UTILIZANDO ISPTV .....</b>	<b>44</b>
4.1 Aplicações Educacionais.....	44
4.2 Aplicações de Realidade Virtual.....	47
4.3 Aplicações Interativas para Inclusão de pessoas com necessidades especiais .	49
<b>5. VALIDAÇÃO DE USABILIDADE E DESEMPENHO .....</b>	<b>52</b>
5.1 Teste de usabilidade .....	53
5.1.1 Implementações da API <i>ispTV</i> .....	54

5.1.2 Teste de usabilidade da API <i>ispTV</i> .....	57
5.2 Teste de tempo de resposta.....	58
5.3 Teste de escalabilidade.....	62
5.4 Considerações Finais .....	65
<b>6. CONCLUSÃO .....</b>	<b>67</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>60</b>

# Capítulo 1

## INTRODUÇÃO

---

---

*Neste capítulo será feita uma breve introdução sobre Televisão Digital e Televisão Digital Interativa, as motivações, objetivos e organização deste trabalho.*

### 1.1 Considerações Iniciais

Igualmente a outros meios de comunicação em massa, a televisão não foge da tendência de evolução tecnológico-digital. Com essa evolução, o sistema de televisão analógico está em processo de migração para um novo cenário, que cada vez mais está presente em território nacional. Tal cenário engloba modificações relevantes à forma de produção de conteúdo, envolvendo a produção e a criação de programas digitalmente com alta qualidade de áudio/vídeo. Esta maneira inovadora de se fazer televisão chama-se Televisão Digital (TVD) e, com sua constante evolução uma vasta gama de serviços serão agregados a programas e transmitidos aos telespectadores.

Para que serviços originados pela utilização da TVD continuem a evoluir e se tornem realidade em âmbito nacional e no Sistema Brasileiro de Televisão Digital (SBTVD), o Brasil tem adotado como objetivo instituir esta tecnologia para possibilitar a inclusão social e tecnológica no país, agregando características de multiprogramação e transmissão de dados junto aos programas televisivos. Com essas características, o SBTVD oferece aos telespectadores a possibilidade de interação com os programas televisivos, nos quais os dados se tornam aplicações

interativas e passam a incorporar características de sistemas de Televisão Digital Interativa (TVDI), tornando os telespectadores não mais passivos diante dos programas, mas, usuários dos programas interativos.

Os programas televisivos são comuns, produzidos pela emissora e sem conteúdo interativo. Uma vez que a interatividade está presente, a responsável por ela são as aplicações interativas. Estas possuem toda a interatividade associada a um programa e, quando estão associados, as emissoras passam a produzir programas interativos (programa televisivo mais aplicação interativa) e não apenas programas televisivos.

A televisão, em âmbito nacional, é o meio de disseminação de informação mais utilizado, atingindo 95,1% da população, com superação aos computadores, que atingem cerca de 31,2% da população e da Internet, que atinge 23,8% (IBGE; PNAD, 2009). A televisão digital interativa tem como objetivo oferecer aos telespectadores uma nova forma de disseminação da informação, possibilitando a agregação de aplicações com interatividade, presente na Internet, aos programas televisivos. Através da interatividade, a televisão digital interativa possibilita a produção de serviços personalizados e interativos através das aplicações (FERNANDES *et al.*, 2004). Com isso, o papel do usuário na produção, transmissão e consumo do conteúdo e a forma com que os usuários interagem entre si, diversificando com o gênero dos programas, é de importância para as emissoras obterem os perfis de cada usuário que interage com seus programas televisivos. (GEERTS. *et al.*, 2008) (CESAR. *et al.*, 2008).

Neste sentido, a TVDI é, na verdade, um universo complexo e cheio de possibilidades envolvendo a criação de programas interativos, o que caracteriza uma nova cadeia produtiva, como planejamento e desenvolvimento, especialmente em termos de estratégias a serem adotadas para prover interatividade aos usuários, fazem parte deste universo. (FERNANDES *et al.*, 2004). É neste cenário que este trabalho se enquadra e encontra suas motivações e objetivos, inseridos no contexto de prover interatividade aos usuários de sistemas de TVDI.

## 1.2 Motivação

De acordo com Decreto Brasileiro nº 4.901, o objetivo principal do Governo Brasileiro com a adoção do SBTVD é “promover a inclusão social, a diversidade cultural do País e a língua pátria, por meio do acesso à tecnologia digital, visando à democratização da informação” (Decreto N° 4.901, 2003).

Para promover a inclusão social, o SBTVD deve possibilitar a incorporação de aplicações interativas aos programas televisivos por meio do desenvolvimento de aplicações interativas por parte das emissoras e a incorporação aos programas televisivos permitindo ao usuário o acesso a interatividade durante a exibição dos programas televisivos.

Os serviços governamentais, de educação à distância e de comércio eletrônico são utilizados na Internet para facilitar o acesso às informações disponíveis na web, sendo nomeados como serviços *e-government*, *e-learning* e *e-commerce*. Com a expansão da TVDI no país, as emissoras de televisão, com base no Decreto Brasileiro, podem produzir aplicações interativas que ofereçam as mesmas características disponíveis na Internet, diferenciando-se pela forma de acesso, que é a televisão. Estas aplicações, levando em consideração a classificação da Internet, recebem a classificação de *t-government*, *t-learning*, e *t-commerce*. Para viabilizar esse cenário, a interatividade deve estar presente no SBTVD, onde Provedores de Serviços Interativos (PSI) possam ser utilizados como provedores de informações, referentes aos programas e aplicações interativos, possibilitando a interatividade e possibilitando a inclusão social e tecnológica no país.

Neste sentido, a motivação deste trabalho é oferecer às emissoras a possibilidade de desenvolverem aplicações interativas para sistemas de TVDI, utilizem todas as características de comunicação oferecida pelo canal de interatividade com ferramentas que facilitem a comunicação com o PSI para obter e publicar novas informações através de serviços web. Com isso, garantir uma arquitetura que possa ser utilizada por pequenas emissoras, emissoras independentes, instituições de pesquisa e desenvolvedores de aplicações interativas de forma a suportar a quantidade de telespectadores interagindo diante um

programa televisivo e oferecendo aplicações interativas com o objetivo de inclusão social e tecnológica.

A motivação também foca em viabilizar o desenvolvimento tecnológico, envolvendo produção de conteúdo e programas televisivos através de tecnologias padronizadas, e em possibilitar a disseminação da informação com o intuito de inclusão social e tecnológica utilizando o *middleware* brasileiro Ginga. Desta forma as emissoras se concentram na produção de conteúdo e deixam de se concentrar também no desenvolvimento de tecnologias de comunicação do canal de interatividade.

A motivação secundária deste trabalho é demonstrar o que a arquitetura proposta agrega novos conceitos e tecnologias diante os trabalhos relacionados já existentes em âmbito nacional e internacional e ao SBTVD.

### 1.3 Objetivos

Diante a motivação apresentada, este trabalho apresenta a arquitetura de um PSI para sistemas de TVDI para ser utilizada de forma eficaz nas classes de sistemas de TVDI destacados na motivação que ofereçam suporte ao canal de interatividade. O objetivo deste trabalho é oferecer uma arquitetura genérica e padronizada – *Interactive Service Provider for Digital Television Systems - ispTV* – de um PSI, com base em uma Arquitetura Orientada a Serviços (SOA), garantindo aos desenvolvedores de aplicações interativas e serviços web maneiras simplificadas de comunicação com o PSI e de oferecimento de serviços para serem acessados pelas aplicações interativas.

O objetivo de desenvolver uma arquitetura de um PSI está relacionado à facilidade de publicação de conteúdo pelas emissoras referentes às aplicações interativas desenvolvidas, à facilidade de desenvolvimento das formas de comunicação com estes serviços e a interação do usuário com as aplicações interativas.

O objetivo descrito tem foco em oferecer às pequenas emissoras, emissoras independentes, instituições de pesquisa e desenvolvedores de aplicações

interativas, que necessitem de uma arquitetura de canal de interatividade, ferramentas para o utilizarem sem necessidade de adquirir ferramentas existentes para transmitir suas aplicações interativas para os usuários ou para desenvolvimento de aplicações interativas. Isso se deve em razão de as emissoras de grande porte utilizarem sistemas de interatividade próprios e estarem desenvolvendo seus sistemas de canal de interatividade de forma a poderem ser utilizados em rede nacional e com suporte a milhões de usuários.

A arquitetura proposta só faz sentido quando existe uma comunicação entre as aplicações interativas e PSI com suporte ao canal de interatividade, o que acontece com os sistemas de TVDI. Com isso, o objetivo deste trabalho também engloba a validação da arquitetura através do desenvolvimento de aplicações interativas, de educação a distância e inclusão social de pessoas com necessidades especiais, e através da validação de usabilidade das ferramentas que tornam a arquitetura genérica e padronizada. Dessa forma, este trabalho contribui para que as aplicações interativas presentes nos programas televisivos brasileiros possam atender objetivos governamentais. O decreto nº 4.901 descreve a intenção de inclusão social e tecnológica através da televisão, o que permite migrar aplicações, já presentes na Internet, para os programas televisivos. Aplicações educacionais, governamentais e comerciais são as mais exploradas no contexto brasileiro e aplicações educacionais serão apresentadas neste trabalho como prova de conceito para a arquitetura proposta.

Vale observar que o sucesso do SBTVD está diretamente relacionado com o sucesso do uso do canal de interatividade que está diretamente ligado ao oferecimento de interatividade aos telespectadores (FERNANDES *et al.*, 2004).

## 1.4 Organização do Trabalho

A apresentação da arquitetura de um PSI para sistemas de televisão digital interativa está organizada e estruturada em seis capítulos. O primeiro capítulo, no qual esta sessão está inserida, apresenta as motivações, objetivos e organização do

trabalho.

O capítulo 2 explica o que são aplicações interativas e o que é necessário nos sistemas de televisão digital para sua transmissão. Também é apresentado aonde elas podem ser utilizadas para promover a inclusão social e tecnológica.

O capítulo 3 sugere a proposta e o desenvolvimento da arquitetura de um PSI para sistemas de televisão digital e a comparação com outras arquiteturas existentes.

No capítulo 4, são apresentadas as descrições das implementações de aplicações que atenda o decreto governamental utilizando-se a arquitetura proposta neste trabalho.

No capítulo 5, são detalhados os resultados envolvendo testes de comunicação e desempenho relacionados ao funcionamento da arquitetura.

No capítulo 6, apresenta-se a conclusão diante as considerações de cada capítulo deste trabalho, fazendo assim, um desfecho geral de todo o trabalho.

# Capítulo 2

## APLICAÇÕES INTERATIVAS EM TELEVISÃO DIGITAL INTERATIVA

---

*Este capítulo apresenta conceitos sobre aplicações interativas. Serão detalhados os tipos de aplicações interativas e os componentes que estão diretamente ligados a transmissão, execução, interação das aplicações e definições que são adotadas neste trabalho.*

Tratando-se de televisão digital em cenário mundial, existem diversos sistemas que oferecem diferentes maneiras de promover conteúdo digital. Os sistemas mais desenvolvidos são o europeu *Digital Video Broadcasting (DVB)*, o americano *Advanced Television System Committee (ATSC)*, o Japonês *Integrated Services Digital Broadcasting - Terrestrial (ISDB-T)*, e o brasileiro *Integrated Services Digital Broadcasting – Terrestrial Brazilian (ISDB-TB)*. As características de cada sistema estão diretamente relacionadas com os interesses de cada país ou região em relação à televisão digital, podendo predominar a qualidade de imagem ou a diversidade de aplicações interativas.

O sistema europeu DVB tem como objetivo a transmissão de programas com maior qualidade de áudio/vídeo e com possibilidade de adicionar aplicações interativas. Tal sistema utiliza o *middleware Multimedia Home Platform (MHP)* (DVB, 2009) que oferece APIs para a criação de programas interativos.

O sistema americano ATSC objetiva a transmissão dos programas televisivos com maior qualidade de áudio/vídeo possível, sem a intenção de oferecer a possibilidade de interatividade para o telespectador. O middleware utilizado por esse sistema é o *DASE* (ATSC, 2009) e possibilita que aplicações interativas sejam agregadas ao sistema ATSC mesmo não sendo o objetivo do sistema.

O sistema japonês ISDB-T, base para o Sistema Brasileiro de Televisão Digital, tem como finalidade principal a transmissão de programas, com foco na diversidade

de aplicações interativas. A portabilidade e mobilidade são características deste sistema, objetivando a transmissão para dispositivos móveis. O middleware utilizado pelo sistema ISDB-T é a *Association of Radio Industries and Businesses* (ARIB) (ISDB-T, 2009) e centra-se na transmissão de aplicações interativas junto aos programas televisivos.

As características comuns dos *middlewares* citados acima estão ligadas à possibilidade de oferecimento de conteúdo interativo junto à exibição dos programas televisivos. A utilização de cada *middleware* se difere diante as intenções de cada sistema de televisão digital.

O Sistema Brasileiro de Televisão Digital (SBTVD) utiliza o sistema japonês como base, com projeto iniciado em novembro de 2003 pelo Decreto Brasileiro de nº 4.901, que oferece à ciência nacional a possibilidade de pesquisa e de desenvolvimento tecnológico com a intenção de promover a inclusão social e cultural no país (Decreto N° 4.901, 2003).

Ao observar as necessidade e condições específicas para a implantação do Sistema Brasileiro de Televisão Digital, o padrão Japonês (ISDB) foi escolhido como base para utilização da parte referente à transmissão e à recepção, que são referentes às camadas inferiores do sistema. Para as camadas superiores, *middleware* e aplicações, especificou-se que seriam desenvolvidas pelo Brasil para garantir todas as características determinadas pelo decreto nº 4.901 (Decreto N° 4.901, 2003). Com o conjunto de tecnologias japonesas e brasileiras, deu-se origem ao ISDTV-T ou SBTVD.

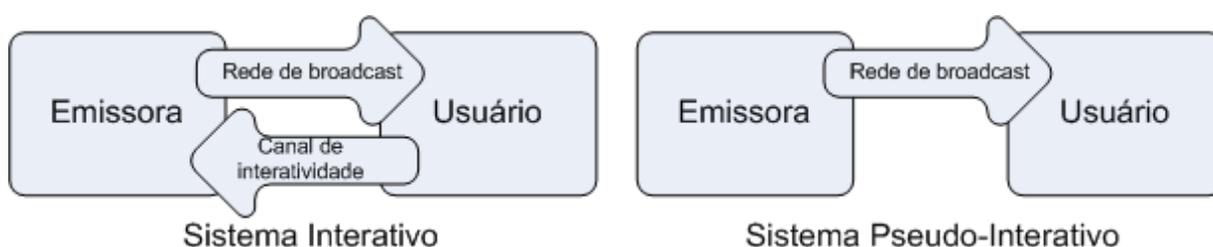
As inovações brasileiras para o SBTVD estão concentradas na camada de software, na qual, inclui o *middleware* e suporte a aplicações interativas. A especificação do *middleware* brasileiro, nomeado Ginga, oferece às aplicações a abstração das diferentes tecnologias utilizadas no hardware e sistema operacional presente nos *Set-Top Box* (STB). Com esta abstração, o suporte a diferentes tipos de aplicações interativas é oferecido por duas implementações do *middleware*, em Java e NCL, e uma ponto entre ambos os ambientes que garante o suporte a aplicações híbridas, podendo executar aplicações Java e NCL juntas (GINGA, 2010).

O SBTVD oferece às emissoras a capacidade de produção de programas interativos com diversas classes de aplicações interativas sendo elaboradas e desenvolvidas para serem transmitidas para os telespectadores.

## 2.1 Aplicações Interativas

Com a possibilidade de produção de aplicações interativas, surge uma nova linha de produção envolvendo programas televisivos com aplicações interativas (programas interativos), interesses governamentais e foco no usuário. A possibilidade de interação do usuário com os programas interativos modifica toda a forma de organização de um sistema de televisão, que passa a se preocupar com o ciclo das suas aplicações junto aos programas (MONTEZ; BECKER, 2005).

Um programa televisivo possui um ciclo de vida contínuo, sendo produzido, transmitido e oferecido aos usuários. Quando as aplicações interativas são levadas em consideração, o ciclo de vida dos programas interativos continua a existir mas com classificações diferentes envolvendo o ciclo de vida de suas aplicações. O ciclo de vida das aplicações interativas classificam os sistemas de TVDI em sistemas pseudo-interativo ou interativo, ilustrados pela figura 1.



**Figura 1:** Sistema Interativo e Pseudo-Interativo

A principal diferença de classificação dos sistemas interativos e pseudo-interativos está na utilização, ou não, de um canal de interatividade. Um sistema pseudointerativo de televisão possui o ciclo de vida de uma aplicação interativa limitado a troca local de informações restringindo-se a subsistemas de recepção doméstica (STB). Com isso, estes sistemas podem apenas armazenar e obter

dados, utilizando o *middleware* ou demonstrar conteúdo para os usuários sem formas de comunicação, através do canal de interatividade. Um sistema interativo possui as mesmas características do sistema pseudointerativo, diferenciando-se na possibilidade de comunicação das aplicações interativas com o meio externo via canal de interatividade. Essa característica possibilita à emissora agregar maiores funcionalidades às aplicações interativas tornando possível que dados sejam enviados a partir do STB, presente no usuário, para a emissora para criação e atualização de informações e conteúdos em tempo real. Diante dos dois tipos de sistemas, suas aplicações interativas também são referenciadas como pseudo-interativas e interativas.

Com as aplicações interativas, os programas televisivos podem oferecer diversas classes de aplicações, envolvendo os tipos de aplicações citados acima. As classes de maior foco diante o decreto n° 4.901 são as de *t-government* e *t-learning*, onde as aplicações governamentais tem como característica oferecer ao telespectador funcionalidades encontradas na Internet ou apenas em estabelecimentos governamentais. Possibilitam ainda a oferta de aplicações educacionais, que têm como característica oferecer aos telespectadores diversas formas de aprendizado relacionado desde a programas educativos até a aulas de cursos a distância. O objetivo com tais aplicações é possibilitar que os telespectadores tenham acesso a serviços e informações governamentais de uma forma simples e acessível.

Para a produção, transmissão e execução das aplicações interativas, alguns componentes e funcionalidades presentes nos sistemas de Televisão Digital Interativa são necessários. A seguir serão detalhados alguns componentes que estão presentes no ciclo de execução de uma aplicação interativa e, conseqüentemente, de um programa interativo.

## 2.2 Set-Top Box

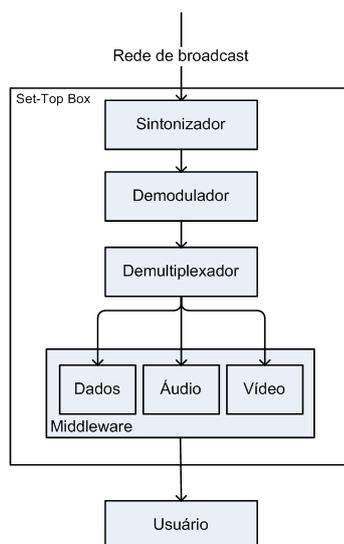
O sinal digital contendo dados dos programas televisivos é recebido pelos usuários, após os fluxos de áudio/vídeo e dados serem codificados, multiplexados, modulados pela emissora de televisão e transmitidos através da rede de broadcast.

Para o recebimento do sinal digital, o usuário necessita de um receptor com as capacidades inversas às executadas pela emissora. Este receptor é chamado de STB (Set-Top Box) e possui este nome por ser um receptor utilizado junto a televisão do telespectador para a obtenção e envio do sinal recebido para o televisor.

O STB possui como característica principal obter o sinal digital enviado pelas emissoras e disponibilizá-lo ao usuário como programas televisivos. Para que isso ocorra, o sinal recebido deve ser primeiramente sintonizado e, em seguida, demodulados, demultiplexados e decodificados para que os fluxos possam ser diferenciados e extraídos para serem entregues ao decodificador exibindo-os em seu correto formato. O processo de sintonia do sinal é responsável por localizar e garantir a correta captação do sinal digital. (MONTEZ; BECKER, 2005).

Relacionadas com a capacidade de envio de dados junto aos fluxos de áudio/vídeo, o STB também possui as características de processamento e armazenamento para que aplicações interativas sejam enviadas junto aos programas televisivos, formando um programa interativo, e executadas quando recebidas. O componente responsável por receber e executar as aplicações interativas é o *middleware*, um componente intermediário com a capacidade de abstrair para os produtores das aplicações as complexidades envolvidas no processo de obtenção do sinal digital e exibição de vídeo/áudio/dados aos usuários. (FERNANDES *et al.*, 2004).

Além das características já citadas, o *middleware* oferece às aplicações interativas a portabilidade para todos os STB que utilizem a mesma especificação de *middleware*. Esta portabilidade é garantida pelas *Application Programming Interface* (APIs) oferecidas que padronizam a forma de acesso às funcionalidades de cada hardware dos STB. A figura 2 ilustra a arquitetura de um STB junto a um *middleware*.

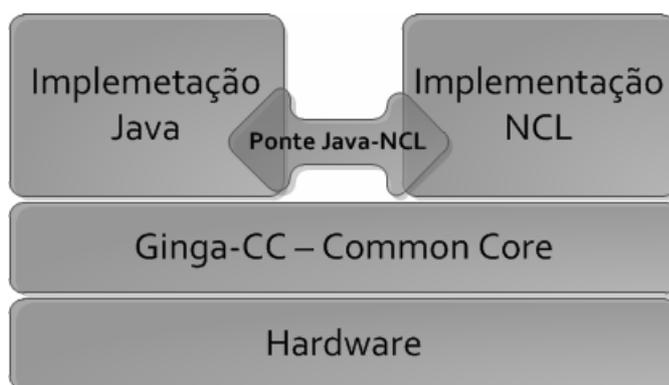


**Figura 2:** Arquitetura Set-Top Box

Devido aos benefícios oferecidos pelo middleware, diversos países e seus órgãos responsáveis pelos padrões de televisão digital, propõem a utilização desta camada em seus sistemas.

No contexto do SBTVD, o *middleware* utilizado e desenvolvido é o Ginga-NCL, que possui duas implementações, denominadas Ginga-NCL e Ginga-JAVA, para a execução e gerenciamento das aplicações interativas.

O ambiente declarativo Ginga-NCL, ilustrado pela figura 3, oferece a possibilidade do desenvolvimento de aplicações interativas utilizando a linguagem declarativa *Nested Context Language* (NCL) e a sua linguagem de script Lua (GINGA, 2010). Já o ambiente imperativo Ginga-JAVA, oferece ao desenvolvedor o suporte ao desenvolvimento de aplicações interativas utilizando a linguagem de programação Java (Xlets), possibilitando o desenvolvimento de aplicações em linguagem procedural. Para a execução das aplicações híbridas, que utilizam os dois ambientes, é oferecida uma ponte entre os ambientes que garante esta capacidade. Ao executar uma aplicação interativa, na qual o Ginga-JAVA é a implementação utilizada, os elementos NCL podem ser modificados através das APIs ofertadas pela implementação desta ponte. A execução de uma aplicação interativa utilizando o Ginga-NCL, possibilita que as aplicações interativas Java sejam interpretadas como elementos de mídia NCL.



**Figura 3:** Arquitetura Middleware Ginga - adaptado de (GINGA, 2010)

Os ambientes Ginga-NCL e Ginga- JAVA estão localizados em uma camada superior ao Núcleo Comum do *middleware* Ginga, o Ginga-CC (GINGA, 2010). O Ginga-CC oferece diversas funcionalidades comuns aos ambientes para execução de suas aplicações interativas. A abstração do hardware e do sistema operacional é garantida pelo Ginga-CC através de funcionalidades comuns responsáveis pela sintonização dos programas televisivos, exibição dos programas interativos e suporte a canal de interatividade.

Uma das partes que está ligada aos *middlewares* e pode possuir componentes, que funcionam com comunicação externa e assim adicionando mais interatividade aos sistemas de televisão digital, é o canal de interatividade, detalhado a seguir.

## 2.3 Canal de Interatividade

A grande inovação obtida com a adoção dos sistemas de televisão digital interativos está relacionada a possibilidade de transmissão de dados (aplicações interativa) junto aos programas televisivos e a possibilidade de interação dos usuários com os aplicativos interativos originados destes dados transmitidos. Com isso, possibilita às emissoras produzirem conteúdo em tempo real baseando-se nas pesquisas, votações, quiz, dentre outros dados enviados pelos usuários.

Para que essa forma de comunicação entre emissora e usuário esteja disponível para os sistemas de TVDI, os sistemas de televisão digital devem garantir

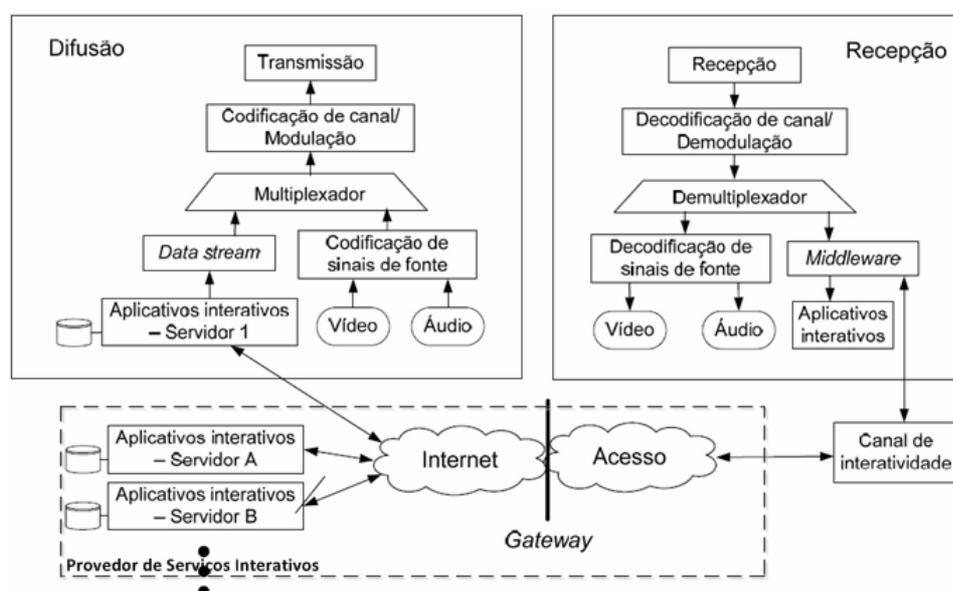
meios para viabilizar a criação de uma arquitetura de canal de interatividade que esteja localizada entre emissoras e usuários, passando a serem classificados como um sistema de TVDI (MANHÃES; Shieh, 2005).

O canal de interatividade tem como objetivo interligar emissoras e usuários para garantir uma comunicação bidirecional com a finalidade, por parte dos usuários, de enviar dados referentes às aplicações interativas para as emissoras, e obter respostas e serviços referentes às aplicações interativas com as quais estão interagindo e, por parte da emissora, a utilização do canal de interatividade tem como objetivo a obtenção de informações necessárias para oferecer conteúdo atualizado para os usuários na forma de novas aplicações interativas ou de novas informações nos programas interativos.

Para que a emissora possa oferecer estes serviços para os usuários, um PSI deve ser adicionado a esta arquitetura no canal de interatividade (MANHÃES; Shieh, 2005). Um PSI tem como objetivo oferecer diversos serviços para que as emissoras e os usuários possam armazenar dados e obter informações quando necessário. Nos sistemas de televisão digital interativa, o PSI pode estar localizado na emissora ou ser uma empresa contratada para prestar tal serviço. No caso do SBTVD, o PSI estará presente nas emissoras de televisão (MANHÃES; Shieh, 2005)..

Para que os serviços estejam disponíveis e funcionem corretamente em âmbito nacional, a arquitetura de um canal de interatividade deve ser elaborada levando em consideração as capacidades físicas e tecnologias presentes em território nacional.

Vários componentes estão envolvidos na arquitetura de um canal de interatividade e estão presentes em outras partes de um sistema de TVDI, as quais são: emissora e recepção. Assim, definindo uma arquitetura completa com usuário, PSI e emissora que é representada pela figura 4 (ABNT NBR 15607-1, 2009).



**Figura 4:** Arquitetura de um Sistema de Televisão Digital Interativa com Canal de Interatividade adaptado de (ABNT NBR 15607-1, 2009)

A emissora é responsável por produzir os programas interativos, desde sua elaboração e produção dos programas televisivos até o desenvolvimento das aplicações interativas para serem transmitidos junto como um programa interativo. Assim que os programas interativos estão produzidos, o próximo passo é transmiti-los para os usuários via *broadcaster*. Para ser transmitido, deve ser feita uma junção de áudio/vídeo/dados em um único fluxo de transmissão – multiplexação – e ser codificado para a forma de transmissão adotada pelo sistema de TVDI em questão. Outra característica da emissora é possuir componentes responsáveis por obter acesso as informações presentes disponíveis através do canal de interatividade e, ou PSI, e a partir delas produzir ou atualizar informações para um futuro acesso dos usuários através dos programas interativos (MONTEZ; BECKER, 2005).

No usuário, o programa interativo recebido necessita de um tratamento inverso ao aplicado na emissora, sendo decodificado e demultiplexado para o correto tratamento pelo *middleware*. Os programas interativos que necessitam da capacidade de comunicação com o meio externo, o PSI, para que os dados referentes à interação do usuário com um programa interativo sejam enviados e armazenados, do canal de interatividade para que a comunicação com o meio externo seja oferecida a eles. Para que um sistema de TVDI possibilite esta comunicação, um conjunto de funcionalidades deve estar disponível no *middleware*

para que o desenvolvedor de aplicações interativas adicione à sua aplicação interativa esta característica.

Na parte intermediária desta arquitetura, o PSI pode ser considerado uma das partes mais importantes que podem estar presentes no canal de interatividade. Isso pela presença de todos os dados, informações e serviços relacionados a um programa interativo. Um PSI, para gerar informações e oferecer serviços, deve possuir a capacidade de armazenar os dados recebidos e de utilizar esses dados para oferecer os serviços e informações que são solicitados pela emissora ou pelas aplicações interativas. Com isso, o PSI oferece uma comunicação bidirecional tanto no sentido PSI/Emissora quanto PSI/Usuário.

O funcionamento de um canal de interatividade nos sistemas de TVDI necessita de modificações significativas em várias partes do sistema. Estas modificações podem variar bastante de um sistema para outro dependendo de suas características, interesses governamentais e limitações físicas.

No caso do Brasil, a norma brasileira ABNT NBR 15607 (ABNT NBR 15607-1, 2009) especifica que o canal de interatividade deve ser implementado utilizando o acesso à Internet de mais baixo custo disponível em cada região do país. Isso implica que a arquitetura geral do sistema não deve se restringir a um único meio de acesso a Internet, mas ao mais viável em cada região. Vários meios de comunicação estão dentro das possibilidades de uso no padrão brasileiro. Levando-se em consideração a aplicabilidade das aplicações interativas e a disponibilidade tecnológica, os meios mais utilizados são (MONTEZ; BECKER, 2005) (CAMPISTA *et al.*, 2007) Telefonia fixa, Wimax, Banda larga, *Power Line Communications* (PLC) e Internet.

Com as tecnologias citadas neste capítulo, surge a possibilidade de desenvolvimento de arquiteturas que oferecem suporte ao desenvolvimento de aplicações e serviços interativos para facilitar a produção de programas interativos para as emissoras. Neste sentido, o próximo capítulo apresenta a arquitetura de um provedor de serviços interativos que utiliza o canal de interatividade para oferecer as emissoras a possibilidade de transmitir programas com interatividade onde necessitam de transmissão de informações para atualização de conteúdo.

## 2.4 Web services

Em sistemas de TVDI que possuem o canal de interatividade e utilizam internet como possibilidade de canal de interatividade, o Web service é uma opção para atender as características necessárias para as emissoras disponibilizarem serviços que atendam a demanda das aplicações interativas utilizando a comunicação com o meio externo (canal de interatividade) para obter e enviar informações para as emissoras.

A necessidade da oferta de variados serviços e a popularização da internet criam a tendência de oferecimento serviços de forma padronizada, flexível e extensível com a finalidade de simplificar a relação entre clientes e servidores. Esta relação de interoperabilidade dos sistemas pode ser simplificada com a utilização dos Web services.

Web service é um framework com a capacidade de garantir aos desenvolvedores formas simplificadas e padronizadas de oferecer serviços aos usuários finais (aplicações web, *mobile*, TV digital, dentre outras).

Para disponibilizar informações aos usuários finais, o framework utiliza orientação a serviços (ERL, 2005) para descrever negócios e problemas a serem resolvidos do mundo real. A maneira como esses serviços serão disponibilizados e acessados pelos usuários finais depende dos papéis assumidos por cada Web service presente em uma arquitetura e de qual o objetivo ou modelo a ser atribuído a esta arquitetura. Os papéis que podem ser assumidos pelos Web services, demonstrados na figura 5 e 6, são identificados como *service provider*, *service requestor*, *intermediaries*, *initial sender* e *ultimate receiver*.

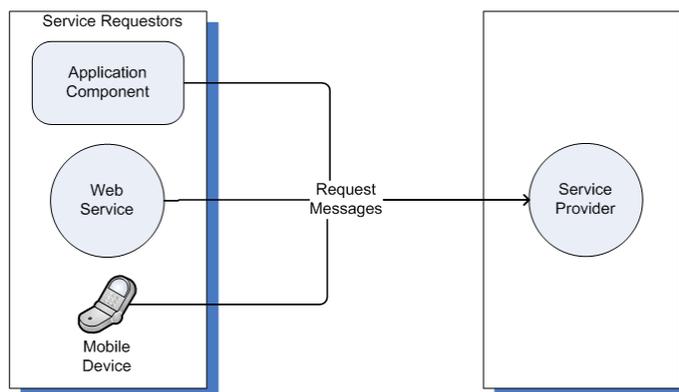


Figura 5 - Requisitores de serviços e provedores de serviços - adaptado de (ERL, 2005)

Um Web service assume o papel de *service provider* (provedor de serviços) ao receber requisições de serviços através de elementos externos, comparando-se ao servidor de uma comunicação cliente/servidor onde dados são requisitados por um cliente e informações são retornadas pelo servidor.

*Service requestors* (requisitores de serviços) são considerados todos os tipos de elementos externos que requisitam um serviço que possa ser entendido por um Web service com papel de *service provider*. Uma aplicação Web, Mobile ou mesmo um Web service pode ser considerado um solicitante de serviço, comparando-se em uma arquitetura cliente/servidor ao cliente solicitando informações a um servidor.

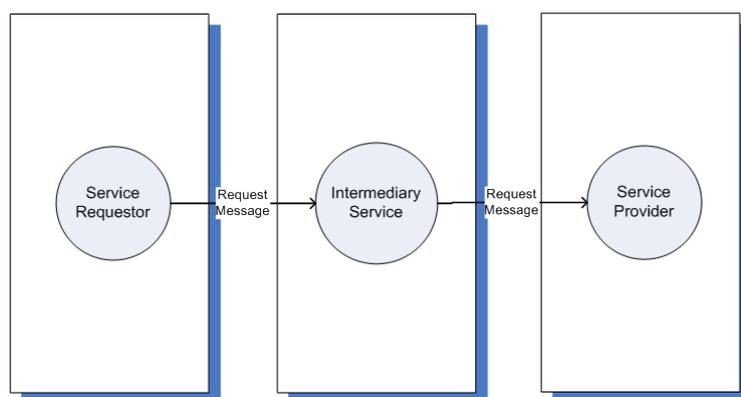


Figura 6 - Papéis de Web services - adaptado de (ERL, 2005)

Pela grande flexibilidade e escalabilidade, os Web services podem assumir outros papéis que não limitam o framework à comunicação cliente/servidor. Com isto, outra característica é a possibilidade de Web services com papéis de *intermediaries* (intermediários). Eles podem trabalhar como roteadores ou

processadores de mensagens antes que cheguem a seu destino, podendo ser classificados em intermediário passivo e intermediário ativo. A classificação em intermediários passivos deve-se à não modificação das mensagens transmitidas após a interpretação de seu cabeçalho. Os intermediários ativos diferenciam-se dos intermediários passivos pela modificação dos dados presentes no cabeçalho e no conteúdo da mensagem.

Os Web services e elementos externos podem ser classificados como requisitores de serviços. Quando o ponto inicial da transmissão da mensagem parte de um destes elementos, eles são classificados como *initial senders* (remetentes iniciais). Ao contrário dos remetentes iniciais, os *ultimate receivers* (receptores finais) são provedores de serviços considerados como o último receptor da mensagem enviada.

Com a grande variedade de papéis exercidos pelos Web services e a necessidade de oferecimento de serviços para aplicações, os diferentes papéis podem ser utilizados para comporem serviços específicos. A composição de serviços não é um termo utilizado para descrever um papel específico de um Web service, mas se refere a um conjunto de Web services com o objetivo final de cumprir uma tarefa. Para que este conjunto cumpra uma tarefa, sua tarefa principal é dividida em sub-tarefas que são específicas de cada Web service. Cada Web service presente é classificado como um *composition member* (membro da composição) do serviço, pois são responsáveis por pequenas tarefas ou tarefas específicas relacionadas à tarefa principal.

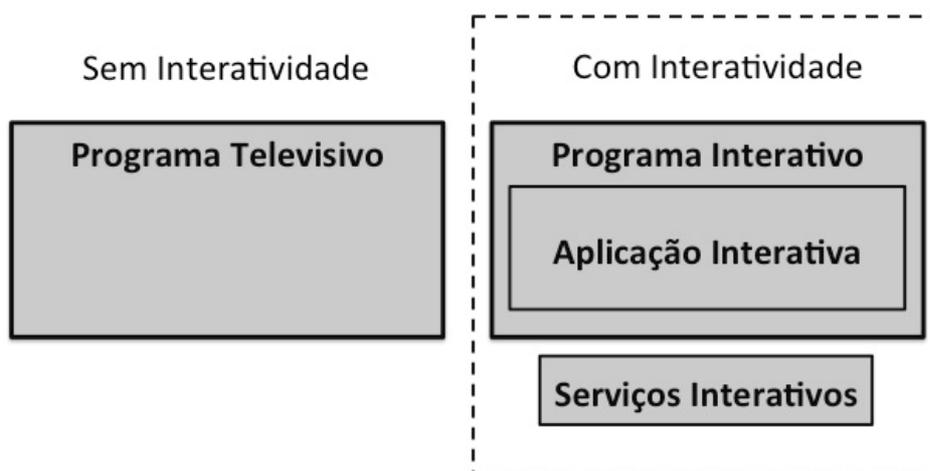
Com a composição de serviços, partindo da união de diferentes papéis dos Web services, vários modelos podem ser propostos a partir das composições de serviços de Web services. Alguns dos serviços mais comuns que são visualizados no mundo real podem ser classificados em modelo de negócios, utilidades e controle. O modelo de negócios representa conjuntos de lógicas de negócios, geralmente sendo utilizadas em composições de serviços. Para representar situações com funcionalidades genéricas, o modelo de serviço utilizado é o de utilidades. Já o modelo de controle é utilizado pelos Web services responsáveis por controlar composições de serviços.

Para obterem o fraco acoplamento (menor dependência entre Web services) em suas comunicações, Web services utilizam documentos de descrição de serviços para a comunicação entre requisitores de serviços e provedores de serviços.

O documento de descrição de serviços, nomeado WSDL, tem como objetivo descrever *endpoints* (destinatário das mensagens) oferecendo aos requisitores de serviços descrições lógicas e físicas dos serviços disponíveis possibilitando o roteamento da mensagem. Geralmente os *endpoints* são representados pelos Web services provedores de serviços.

Para descrever os serviços presentes nos *endpoints*, um documento WSDL é dividido em uma parte de descrição abstrata e em outra de descrição concreta. A descrição abstrata de um documento WSDL (*Web Service Definition Language*) contém informações referentes à interface do Web service, descrevendo seus serviços e suas respectivas funcionalidades e parâmetros de entrada e saída. Diferentemente da descrição abstrata, a descrição concreta se concentra na descrição física das tecnologias a serem utilizadas para a execução das lógicas presentes nos Web services e para que a interface abstrata possua referências físicas de um *endpoint*.

Com os conceitos apresentados sobre TVDI e Web services, este trabalho apresenta alguns conceitos, representados na figura 7, sobre programas televisivos e a relação com as aplicações interativas, tanto na transmissão quanto na recepção:



**Figura 7** – Conceitos de interatividade referente a programas televisivos e aplicações interativas.

A base do funcionamento de toda a interatividade é o programa televisivo, o qual é representado por áudio e vídeo e nada mais é do que os programas que são transmitidos para os usuário em todo o território nacional. Já um programa interativo representa a agregação das características de interatividade aos programas televisivos, em que são inseridas as aplicações interativas e seu respectivos serviços interativos. As aplicações interativas, já descritas, representam toda forma de interatividade com o usuário oferecida pelos programas interativos e os serviços interativos, classificados como interativos pela associação com as aplicações, representam as fontes de obtenção e armazenamento de informações para as aplicações oferecendo mais riqueza para a interação com os usuários.

Com isso, ao transmitir um programa interativo em rede nacional, seria transmitindo o áudio e vídeo de um programa televisivo mais a capacidade de interatividade associada ao programa. Sendo assim, a arquitetura a ser proposta trabalha com estes conceitos para cumprir todos os seus objetivos.

# Capítulo 3

## ARQUITETURA DE UM PROVEDOR DE SERVIÇOS INTERATIVOS

---

*Neste capítulo será apresentado uma Arquitetura de um Provedor de Serviços Interativos, baseada em uma Arquitetura Orientada a Serviços, para sistemas de Televisão Digital Interativa contemplando funcionalidades na emissora, recepção e canal de interatividade.*

As implementações do *middleware* brasileiro Ginga estão em constante evolução, visando ao atendimento das características presentes nas normas brasileiras de televisão digital. Tratando-se da execução das aplicações interativas junto aos programas televisivos (programas interativos), o *middleware* tem se desenvolvendo de forma confiável para que a execução de programas interativos ocorra (SOARES *et AL.*, 2007). No estado atual das implementações do *middleware* Ginga, Ginga-NCL e Ginga-JAVA, o Ginga-NCL é o que possui a implementação das características necessárias para comunicação via canal de interatividade mais bem desenvolvidas. Esta comunicação é feita através de APIs presentes no *middleware* e é oferecida através da linguagem de *script* Lua, utilizada junto ao NCL, para oferecer formas de comunicação via *Internet Protocol* (IP).

Para a produção de conteúdo interativo para o telespectador, as emissoras devem oferecer em suas aplicações interativas formas de comunicação e interação com o meio externo (canal de interatividade). Para que isso ocorra, servidores de informações, mais especificamente PSI, devem ter a capacidade de suporte a diferentes tecnologias de comunicação que podem ser utilizadas nas aplicações interativas produzidas e transmitidas pelas emissoras.

Neste cenário, este trabalho propõe uma arquitetura genérica e padronizada – *Interactive Service Provider for Digital Television Systems - ispTV* – de um PSI com

base em uma SOA que ofereça facilidade de produção de conteúdo pelas emissoras a partir de serviços web e a abstração das formas de comunicação por parte dos desenvolvedores de aplicações interativas com os serviços. A possibilidade de armazenamento das aplicações interativas junto aos seus respectivos serviços web também é oferecida como suporte a organização das aplicações junto aos seus serviços interativos.

Este trabalho tem como base uma prova de conceito desenvolvida (PRADO, G.M.; ZORZO, S.D., 2010) oferecendo serviços organizados apenas localmente, não sendo utilizadas tecnologias para oferecer a genericidade de tecnologias de publicação de serviços e padronização de comunicação, sendo melhorada para garantir melhoras nas formas de publicação de serviços e comunicação genérica via canal de interatividade.

A arquitetura propõe-se oferecer o suporte à comunicação via canal de interatividade para sistemas de TVDI. Por ser uma arquitetura genérica, pode ser utilizada por diversos tipos de aplicações desde que a forma de comunicação seja o protocolo de comunicação oferecido pela arquitetura. Neste trabalho, a arquitetura propõe-se a ser adotada por sistemas de TVDI e utilizada pelo *middleware* brasileiro Ginga para garantir que as aplicações interativas se comuniquem com *Web services*. A arquitetura foi elaborada para oferecer módulos em algumas partes de um sistema de TVDI, que terá funcionalidade na emissora, recepção (STB) e canal de interatividade oferecendo ferramentas nas três partes destacadas dos sistemas de televisão digital.

Os módulos serão detalhados em diferentes sessões e, em seguida, será apresentado seu funcionamento diante a produção/transmissão/execução de um programa interativo.

A seguir é apresentada a arquitetura genérica de um Provedor de Serviços Interativos – ispTV – proposta, como o detalhamento de seu funcionamento de modo geral, sua comunicação com o *Universal Description, Discovery and Integration* (UDDI) da arquitetura e a descrição dos serviços específicos *ispTV*. Será detalhado o funcionamento de sua aplicação web de gerenciamento – ispTV Web Tool (ispTV WT) – para publicação de serviços, sua API cliente de comunicação – ispTV Client (ispTV C) – e a mensagem padrão de comunicação utilizada pelo *ispTV C*. Após as

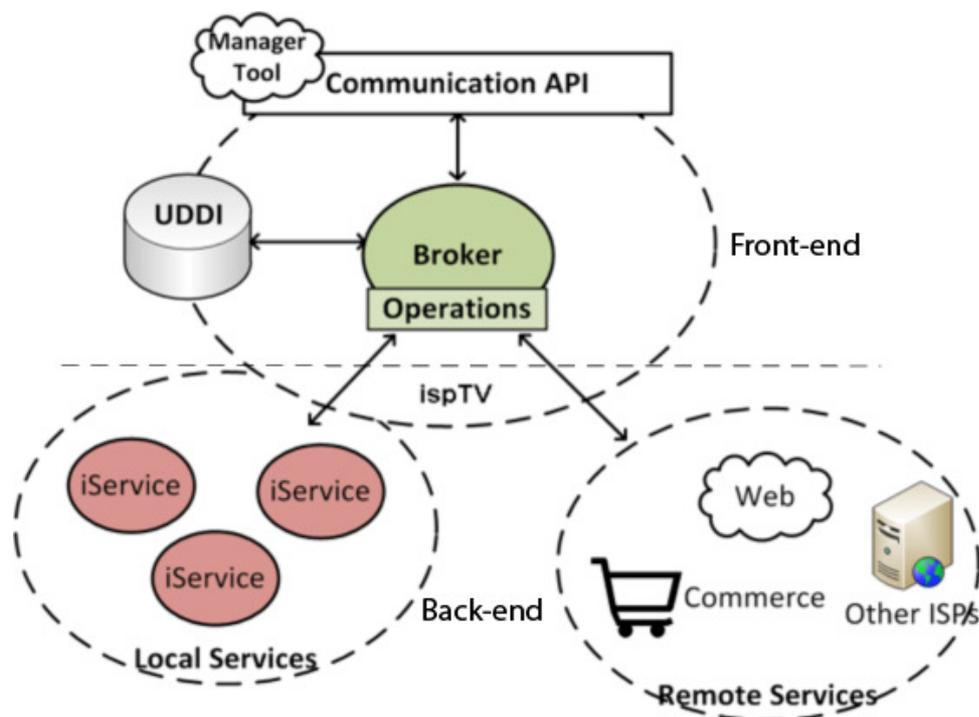
descrições dos módulos que contemplam a arquitetura, será apresentada a sua utilização no contexto brasileiro de televisão digital e trabalhos relacionados a arquitetura *ispTV* com suas contribuições.

### 3.1 Arquitetura Genérica - *ispTV*

A arquitetura genérica do PSI – *ispTV* – possui como ideia principal a concentração de serviços interativos referentes às aplicações interativas com a finalidade de registrar e manter grupos de serviços com propostas comuns de interatividades em um mesmo PSI. A partir deste objetivo surge uma arquitetura genérica, padronizada e expansível, que oferece a garantia de publicação de serviços em diferentes tecnologias com o objetivo de oferecer informações comuns às aplicações interativas transmitidas.

A arquitetura proposta é baseada em uma composição de *Web services* (Serviços Web) para permitir que novos serviços sejam adicionados na arquitetura sem modificar sua estrutura e o modo de acesso aos serviços pelos clientes. Serviços da Web foram adotados por sua independência de aplicação e pelas características genéricas de comunicação (ERL, 2005).

Na arquitetura *ispTV*, os serviços interativos utilizados são organizados através de uma composição de serviços que possuem papéis definidos como intermediários ou prestadores (ERL, 2005). Esses papéis definem duas camadas estruturais para a arquitetura, a camada de *front-end* e a camada de *back-end*. A primeira categoria de serviços – serviço de *front-end* – está localizada entre as aplicações interativas e a camada de *back-end*, que possuem o papel de receber as requisições feitas pelas aplicações interativas e redirecionar para a camada de *back-end*. A camada de *back-end* está relacionada aos serviços interativos acessados pelas aplicações interativas sendo os serviços que possuem as informações utilizadas e publicadas pelas aplicações interativas. Assim, a arquitetura *ispTV* é ilustrada na figura 8 demonstrando as camadas de *front-end* e *back-end*.



**Figura 8:** Arquitetura do módulo do Provedor de Serviços Interativos

A camada de *front-end* é a única forma de acesso oferecida às aplicações interativas pela arquitetura *ispTV*. Esta camada oferece maneiras de publicar e acessar os serviços, cujo o gerenciamento de acesso é feito através de um serviço intermediário – *Broker Service* – que oferece uma interface padrão de comunicação. O *Broker* oferece três funcionalidades (*Post*, *List* e *Route*) responsáveis pela publicação de serviços, lista de serviços e roteamento de requisições. A funcionalidade *Post* (publicação de serviços) é responsável por garantir formas de publicação de serviços específicos das aplicações interativas no *ispTV* para futuro acesso das aplicações interativas e da emissora. Seu funcionamento baseia-se no tipo de serviço a ser publicado, podendo ser classificados em serviços locais e serviços remotos. Para os serviços locais – *ispTV services* – a publicação é baseada na recepção de um arquivo, que possui a lógica do serviço, baseada no seu armazenamento no banco de dados da arquitetura e seu registro/armazenamento em um registro de serviços (UDDI) (ERL, 2005). Para os serviços remotos, presentes em outros PSI ou na Web, o endereço do serviço é armazenado no UDDI para que seja feito o redirecionamento das requisições quando solicitado. Serviços já produzidos na Web podem ser utilizados como serviços das aplicações interativas

reaproveitando os já desenvolvidos. A funcionalidade *List* (listagem de serviços) oferece a lista de serviços registrados no UDDI, sendo eles locais ou remotos, podendo obter informações deles através de uma lista completa de todos os serviços ou apenas de um serviço em específico. Por fim, a operação *Route* (roteamento de serviços) fornece uma interface de comunicação padrão para as aplicações interativas para requisição e persistência de informações nos serviços publicados no UDDI. Quando uma solicitação de serviço é recebida, uma verificação no UDDI é feita para verificação de existência do serviço requisitado. Caso o serviço esteja publicado, a requisição é redirecionada para o serviço para processamento das informações, sendo elas de obtenção ou armazenamento de informações.

O *Broker* não altera a mensagem de requisição e utiliza *Web services* baseados em *RESTful* (MENG, J. et al, 2009) para o recebimento das requisições. *RESTful* foi escolhido pela simplicidade de padronização das mensagens trafegadas entre os clientes (aplicações interativas) e os serviços. *RESTful* é baseado na transmissão de informação em diversos tipos de dados (HTTP, XML, JSON) sem o comprometimento com protocolos, tal como SOAP que é utilizado nos Web Services padrões, tornando a personalização das mensagens/requisições mais flexível.

Com isso, a comunicação com a arquitetura *ispTV* é feita através de mensagens *RESTful* e é personalizada de certa forma que abstrai os desenvolvedores de aplicações de terem conhecimento específico de como é feita toda a configuração de comunicação com a arquitetura. A mensagem e a abstração serão melhores e mais detalhadas, respectivamente, nas seções 3.1.2 e 3.1.4

A camada de *back-end*, referente aos serviços publicados através do *Broker*, é a camada representativa dos serviços com funcionalidades de prestação de serviços onde são responsáveis por o processamento lógico de informações baseados nas requisições das aplicações interativas. Os serviços presentes nesta camada podem ser classificados em três tipos, sendo eles: *ispTV local services* (serviços locais), *JAX-WS services* (JAX-WS, 2011) e *Axis2 services* (Axis2, 2011). Eles podem ser utilizados na arquitetura para fazer publicação de *Web services* na internet. Os serviços publicados através do *Broker* são classificados em serviços locais e serviços remotos, diferenciando-se pelo seu local de oferecimento/publicação.

Os serviços locais, nomeados em *ispTV local services*, recebem essa classificação pois é o único tipo de serviço que é publicado na arquitetura e está fisicamente localizado dentro da *ispTV*. Sua forma de publicação se assemelha com as dos serviços remotos diferenciando apenas pela utilização de um arquivo, modelado especificamente para a arquitetura, ao invés de um endereço do *endpoint* do serviço. A utilização de um arquivo foi adotada para facilitar o desenvolvimento e a manutenção do serviço diante das alterações que uma aplicação interativa pode sofrer. Seu desenvolvimento é baseado em uma classe *Java* (Java, 2011) simples, sem adições de APIs', mais especificamente um arquivo.jar que contém as classes *Java*, o que facilita o desenvolvimento e não necessita de aprendizado de novas APIs e tecnologias para desenvolvimento de web services. Ao ser publicado, os serviços locais são armazenados na arquitetura *ispTV* e possuem um endereço local de acesso que é utilizado como referência para o serviço *Broker* executar o redirecionamento quando requisitado. Para redirecionamento, o *Broker* utiliza o *Reflection*, que é uma funcionalidade nativa do *Java* de manipulação de toda uma classe *Java*. Com isso, os serviços locais, junto as funcionalidades providas pela arquitetura *ispTV*, podem receber requisições Web sem possuírem configurações de *Web services*.

Os serviços remotos, nomeados de acordo com sua tecnologia utilizada (JAX-WS e Axis2), são publicados da mesma forma que o serviço *ispTV*. O que os diferencia é a utilização de um endereço de localização do serviço (*endpoint*) para redirecionamento das requisições feitas para a arquitetura através do *Broker*. Os serviços remotos suportados envolvem as tecnologias JAX-WS e Axis2 de desenvolvimento de serviços e as APIs cliente funcionam da mesma forma para ambas as tecnologias, modificando apenas algumas configurações do cabeçalho da mensagem. As requisições são feitas após a identificação do serviço remoto e da operação a ser requisitada a esse serviço. Após a identificação, a requisição é encaminhada para o serviço interativo solicitado repassando o processamento da informação. Após o processamento da mensagem pelo serviço interativo, uma mensagem de resposta é enviada para o *Broker* redirecionar a resposta para a aplicação interativa que fez a requisição e solicitou o processamento da mensagem.

Para o *Broker*, ambos os serviços possuem o mesmo comportamento devido

a uma camada de comunicação transparente, que se comunica com o *ispTV UDDI* e que verifica a existência dos serviços, e por APIs clientes específicas, por tipo de serviço. O *ispTV UDDI* é uma implementação de um UDDI com apenas as funcionalidades necessárias para esta arquitetura obter o funcionamento desejado de armazenamento e roteamento de serviços. Com isso, a arquitetura *ispTV* com as camadas de *front-end* e *back-end*, ilustrado pela figura 8, oferece uma interface de comunicação padrão com os serviços publicados na camada de *back-end* e que foram classificados como prestadores de serviços que permitindo a implantação/publicação de diferentes serviços. Com isto e com a mensagem padrão de comunicação, as aplicações interativas não precisam se preocupar com as formas de acesso por tipo de serviço publicado, e sim, apenas com a forma padronizada de comunicação oferecida, que será descrita na seção 3.2, e já contemplada pela API *ispTV C* da arquitetura *ispTV*. Isso permite que os serviços adicionados a arquitetura sejam utilizados sem alterar o modo de acesso usado pelas aplicações e que modificações de arquitetura cuja a padronização está na mensagem a ser utilizada e não nos serviços a serem utilizados sejam transparentes para os desenvolvedores.

Analisando o funcionamento da arquitetura, os desenvolvedores de aplicações interativas ficam encarregados de concentrar esforços em duas partes da arquitetura, sendo o desenvolvimento dos serviços específicos de aplicação interativa e a configuração de comunicação com a arquitetura. Com isso, a arquitetura se responsabiliza pelo recebimento das mensagens e roteamento para os serviços específicos de aplicação interativas cadastrados. O roteamento da mensagem se inicia quando o *Broker* recebe uma requisição de tratamento de uma mensagem para um serviço específico. Ao ser requisitado o tratamento, o *Broker* interpreta a mensagem e busca no *ispTV UDDI* informações sobre o serviço solicitado. Sendo encontrado, a mensagem é encaminhada para o serviço específico para tratamento. Caso o serviço não seja encontrado, uma mensagem padrão é retornada para a aplicação interativa. O processo de roteamento é a parte mais importante da arquitetura *ispTV* por possuir a lógica de conversão das mensagens padrões, entre as aplicações interativas e a arquitetura, para as mensagens e protocolos de comunicação adotados pelos serviços locais, JAX-WS e Axis2. Nesse

processo identifica o que enviar e de que forma fazê-lo sem que as informações e o propósito de oferecer a generalidade não se percam na conversão da mensagem padrão da arquitetura para as comunicações específicas.

Para utilizar todos os recursos oferecidos pela arquitetura, os outros dois módulos, *ispTV WT* e *ispTV C*, propostos juntos à arquitetura, devem ser utilizados durante todo o processo de publicação e desenvolvimento dos serviços e aplicações interativas.

## 3.2 Mensagem Padrão

A mensagem padrão – *ispTV Message* –, como citada na seção anterior, é utilizada para comunicação entre as aplicações interativas e a arquitetura *ispTV* sendo essencial para o correto funcionamento da arquitetura proposta. A mensagem padrão tem como base os protocolos de transporte TCP e UDP para que aplicações interativas que necessitem de controle de sincronismo e confirmação de integridade dos dados utilizem a mensagem junto a API – *ispTV C* – que será destacada na seção 3.1.4, de forma a garantir o recebimento das informações pela arquitetura *ispTV*.

Para que a comunicação esteja de acordo com a norma brasileira ABNT NBR 15607 (ABNT NBR 15607-1, 2009) na qual especifica como deve ser feita a comunicação das aplicações interativas que se utiliza do canal de interatividade, um protocolo composto pela mensagem padrão e a API cliente – *ispTV C* – foi estabelecido para que a integridade dos dados e o oferecimento de uma comunicação padrão fosse oferecida.

No contexto das aplicações interativas, a conexão será baseada na norma ABNT NBR 15607 a qual diz que uma aplicação interativa ao utilizar o canal de interatividade deve possuir as seguintes funcionalidades: conexão, transferência de dados e desconexão (ABNT NBR 15607-1, 2009), ou seja, um protocolo de comunicação orientado à conexão. A conexão e desconexão são implementadas pelo *middleware*, implementação Ginga-NCL, notificando os meios físicos para que se conecte e se desconecte a aplicação interativa de um meio externo. No caso de

outras aplicações interativas que requisitem serviços a arquitetura *ispTV*, as formas de conexão e desconexão são implementadas de acordo com a tecnologia utilizada.

Após a etapa de conexão e desconexão estarem definidas, a integridade de transporte dos dados e de execução da aplicação interativa são características importantes da comunicação. As aplicações interativas e a arquitetura *ispTV* podem possuir diferentes interesses com as mensagens utilizadas na comunicação. Algumas necessitarão de confirmações de recebimento dos dados para a conclusão da aplicação, e outras, apenas enviarão os dados e serão finalizadas sem confirmação de recebimento.

O protocolo composto pela mensagem padrão e a API *ispTV C* utiliza mensagens *JavaScript Object Notation* (JSON) para envio das informações e *flags* adicionais relacionadas a sincronismo e criptografia da informação. Através de um *header* instituído na mensagem, a API *ispTV C*, *middleware* ou *ispTV* deverá especificar a necessidade de uma confirmação, ou não, de conclusão da aplicação interativa, se a mensagem é de requisição (sentido aplicações interativas/*ispTV*) ou resposta (sentido *ispTV*/aplicações interativas), o tipo de serviço e se é uma mensagem de término de aplicação interativa.

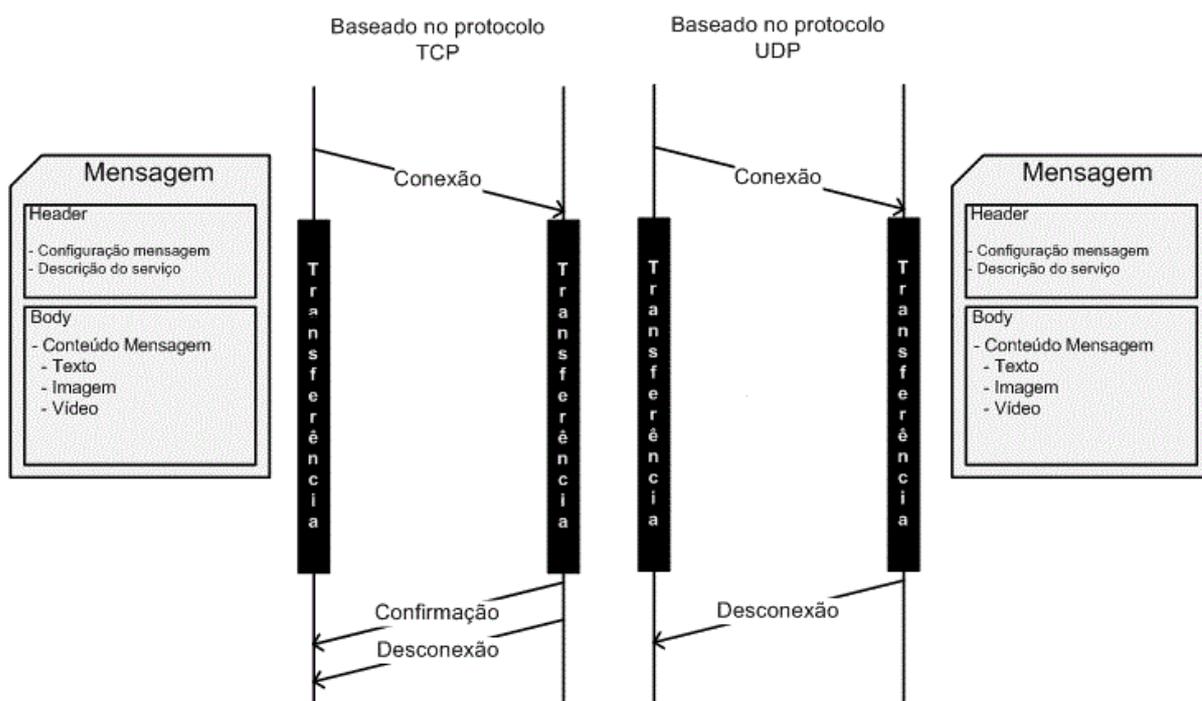


Figura 9: Protocolo com base TCP e UDP

Como ilustrado pela figura 9, a comunicação entre as aplicações interativas e a arquitetura pode utilizar formas de confirmar o término da conexão estabelecida. Uma possibilidade é utilizar a confirmação de término de conexão configurada na mensagem vinda da arquitetura *ispTV* e outra possibilidade é se importar quando uma aplicação deve ser finalizada. Com a confirmação, o protocolo deve ter as características de confirmação habilitadas no *header*, o que possibilita a *ispTV* saber se uma mensagem deve ser retornada à aplicação interativa após receber uma requisição. Caso esteja habilitada, após todas as mensagens entre a aplicação e o *ispTV* serem trocadas, uma mensagem de confirmação é enviada à aplicação notificando o término da comunicação. Por outro lado uma comunicação sem confirmação entre aplicação e a *ispTV* é executada da mesma forma, diferenciando-se pela ausência de uma mensagem de término de aplicação, possibilitando a finalização da aplicação a qualquer momento de sua execução.

```
1 {"header": {
2   "endpoint": "192.168.0.1"
3   "sync": "true",
4   "confirmation": "true",
5   "serviceType": "ispTV",
6   "closeMessage": "false",
7   "content": {
8     // geração recursiva de conteúdo
9   }
10 }}
```

**Figura 10:** Mensagem padrão *ispTV*

A mensagem utilizada na comunicação é ilustrada na figura 10, onde as variáveis que determinam a maneira de comunicação estão representadas da linha 2 até a linha 6 e o conteúdo parte da linha 7. Na linha 3 está configurado se a aplicação interativa deseja enviar uma mensagem síncrona ou assíncrona; Linha 4 está configurado se existe a necessidade de envio de uma resposta de recebimento da mensagem pela arquitetura *ispTV*; A linha 5, especifica o serviço que a aplicação interativa deseja enviar a mensagem; Na linha 6 é especificado se a mensagem enviada é a última a ser enviada pela aplicação, isto quando se sabe realmente qual

é a última mensagem a ser enviada para evitar que conexões fiquem abertas entre a aplicação interativa e a arquitetura; Finalmente, a partir da linha 7 são enviados os dados a serem processados pelo serviço requisitado. Com esta mensagem montada, os desenvolvedores que utilizam o protocolo de comunicação e a API *ispTV C* conseguirão enviar mensagens para os serviços específicos de aplicação registrados na arquitetura *ispTV*.

### 3.3 Web Tool

Para oferecer as funcionalidades de gerenciamento dos serviços, envolvendo publicação, listagem e acesso a informações, a arquitetura *ispTV* oferece uma ferramenta baseada em interesses da emissora chamada *ispTV Web Tool – ispTV WT*. O objetivo da ferramenta *ispTV WT*, ilustrada na figura 11, é oferecer às Emissoras uma forma de gerenciamento para obter informações presentes na arquitetura *ispTV* referentes aos serviços publicados, garantindo o acesso às informações necessárias para o gerenciamento da arquitetura.



The image shows the ispTV Web Tool interface. At the top is the ispTV logo in a maroon rounded rectangle. Below it is a navigation menu with links: Home | Business | Publish | Services | ispTV | Logout. The main form contains the following fields and controls:

- Name:
- Type:
- Service File:  No file chosen
- Description:
- Business:
- 

Figura 11: ispTV Web Tool

Para que a Emissora possa obter as informações armazenadas e produzir novos serviços interativos, formas de acesso e publicação destes serviços foram incorporadas na ferramenta utilizando o acesso direto a arquitetura *ispTV*. A ferramenta *ispTV WT* deve garantir a flexibilidade de atualização dos serviços interativos disponíveis de forma que, ao desenvolver uma nova aplicação interativa, seus serviços estejam disponíveis para acesso. A atualização da ferramenta é feita sempre que um novo serviço interativo é adicionado, atualizado ou removido da arquitetura. Para o desenvolvimento da ferramenta optou-se pela plataforma Java - Java 2 Platform, Enterprise Edition (J2EE) – e um framework J2EE (Struts) para agilizar o processo de desenvolvimento.



Figura 12: ispTV Web Tool

O funcionamento da ferramenta consiste em uma etapa inicial associada à produção de uma aplicação interativa. Em seguida, um serviço interativo relacionado a esta aplicação interativa é produzido com todas suas funcionalidades a serem oferecidas, disponibilizando novas atualizações para a ferramenta e, conseqüentemente, para a aplicação interativa. A última etapa consiste em disponibilizar esta aplicação interativa junto ao seu serviço interativo para a publicação da ferramenta tornando-os disponíveis para manipulação da emissora. Outra característica importante da ferramenta *ispTV WT* é o suporte a publicação das aplicações interativas junto aos seus serviços interativos. Isto para proporcionar à Emissora um acesso único a uma fonte onde estão presentes sua aplicação interativa e seu respectivo serviço interativo. Assim, quando a produção de um programa televisivo necessitar do envio de uma aplicação interativa, a Emissora solicita à aplicação interativa a ferramenta *ispTV WT*, para transmiti-la junto ao

programa televisivo, tornando-o um programa interativo. Os serviços publicados na arquitetura *ispTV* podem ser visualizados na aplicação Web para controle de quais serviços estão publicados, como ilustrado na figura 12.

Com o funcionamento desta ferramenta, a arquitetura *ispTV* oferece às Emissora formas de centralizar suas aplicações interativas e seu serviços em apenas um local, facilitando seu gerenciamento e controle sobre suas aplicações. Reafirma-se assim que serviços já desenvolvidos nas tecnologias JAX-WS e Axis2, também podem ser utilizados como serviços interativos de aplicação junto aos serviços locais *ispTV*.

### 3.4 API Client

As aplicações interativas transmitidas aos usuários são recebidas como dados junto aos fluxos de áudio/vídeo de um programa televisivo. O *middleware* Ginga-NCL é o responsável por identificar os tipos de dados recebidos e garantir sua exibição para os usuários (telespectadores). Uma vez que as aplicações interativas são recebidas seus dados não são mais tratados apenas localmente e necessitam de uma comunicação com um meio externo para seu correto funcionamento, ou seja, com um canal de interatividade e, conseqüentemente, com um PSI, no contexto deste trabalho, com a arquitetura *ispTV*. A forma com que os dados estarão disponíveis para serem enviados para a arquitetura *ispTV* deve ser garantida pelo *middleware* Ginga-NCL ou por APIs desenvolvidas internamente ao *middleware*, presente no STB do usuário, abstraindo as configurações de comunicação do desenvolvedor garantindo uma forma genérica de comunicação. Para garantir esta forma genérica de comunicação, duas APIs cliente – *ispTV Cliente Lua* e *ispTV Cliente Java* – foram desenvolvidas, respectivamente, nas linguagens *Lua* e *Java* para garantir a comunicação utilizando a mensagem padrão/protocolo descrito anteriormente. A API na linguagem *Java* foi desenvolvida para oferecer a aplicação de gerenciamento da arquitetura *ispTV* formas genéricas de comunicação com os serviços de *front-end*, mais especificamente o Broker. Na linguagem *Lua* a API foi desenvolvida independente de APIs do *middleware* com o objetivo de

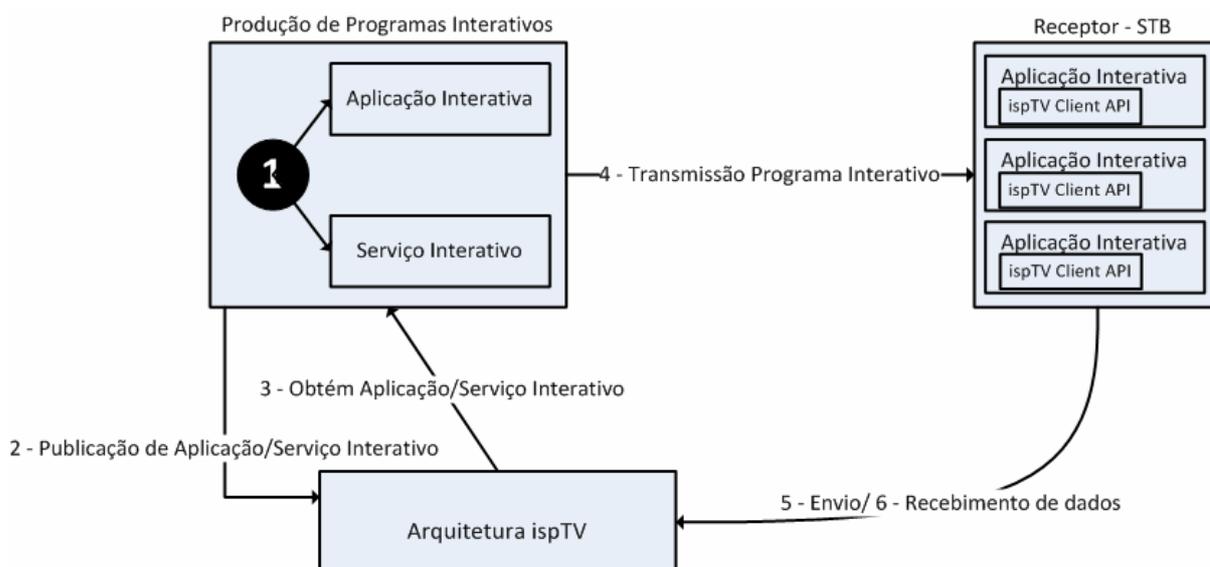
independência de seus módulos para garantir que seu funcionamento seja independente das alterações no core do *middleware*, uma vez que a API poderia ser desenvolvida na Linguagem C, sendo incorporada ao *middleware*, o que tornaria a API muito dependente das versões desenvolvidas do *middleware* Ginga. Desenvolvendo a API em Lua, garante-se que as versões da API sejam totalmente dependentes de suas necessidades uma vez que o suporte a Lua, presente no *middleware*, não interfere no desenvolvimento da API *ispTV C Lua*.

Ambas as APIs assumem que existe o serviço padrão de comunicação *front-end* com a arquitetura *ispTV*, ou seja, tornando customizável o endereço da arquitetura e o serviço a ser requisitado. No caso da utilização desta arquitetura, o serviço padrão a ser utilizado é o Broker.

O objetivo desta API é de garantir ao desenvolvedor de aplicações interativas e de aplicações Java, no caso da aplicação de gerenciamento da arquitetura *ispTV*, que se preocupe apenas com as características da mensagem (síncrona, confirmação de recebimento e última mensagem a ser enviada) e com o seu conteúdo, sem se preocupar com configurações de envio, protocolos e configurações de comunicação com *Web services*.

Logo após uma aplicação interativa entrar em modo de execução e o processo de interação do usuário com a aplicação interativa, pode-se solicitar que o processo de comunicação seja iniciado com o meio externo (arquitetura *ispTV*). Ao iniciar a comunicação, a aplicação interativa obtém acesso aos métodos disponíveis na API utilizando a linguagem Lua determinando um conjunto de atributos relacionados aos dados referentes à configuração da mensagem e ao conteúdo da interação. Após o cabeçalho ser preenchido pela API *ispTV C* (contendo o endereço da arquitetura *ispTV*), os dados da interação são adicionados ao conteúdo da mensagem junto as configurações das características da mensagem. Como processo final da parte de comunicação, a mensagem é enviada para a *ispTV*.

Uma vez que o processo de comunicação foi iniciado, esta API deverá garantir a capacidade de troca de mensagens entre a aplicação interativa e o serviço específico presente no *ispTV* através do protocolo de comunicação a ser proposto.



**Figura 13:** Fluxograma da arquitetura ispTV

O funcionamento geral de utilização de todas as características pelas aplicações interativas da arquitetura é resumido pelo fluxograma na figura 13, no qual as aplicações são desenvolvidas junto ao seu serviço específico (fluxo 1/número 1) e se tornam disponíveis para serem publicadas na arquitetura *ispTV* (fluxo 2/número 2). Ao publicar a aplicação interativa e seu respectivo serviço, informações podem ser obtidas (fluxo 3/número 3) dos dados cadastrados e as aplicações interativas se tornam disponíveis para serem transmitidas e incorporadas (fluxo 4/número 4) aos programas televisivos. Ao ser recebida e processada pelo STB presente no usuário, a aplicação interativa, se existirem dados da interação a ser enviados para a arquitetura *ispTV*, comunica-se (fluxo 5/número 5 e 6/número 6) com a arquitetura *ispTV* para armazenamento e oferecimento dos dados para a emissora em um futuro momento. Esse fluxograma resume como a arquitetura atua em todo o ciclo de uma aplicação interativa transmitida para os usuários.

Com esta seção, pode-se concluir que a arquitetura *ispTV* agrega funcionalidades no desenvolvimento de serviços interativos, com a intenção de facilitar o desenvolvimento destes serviços para as aplicações interativas, e agregar funcionalidades nas formas de comunicação (APIs de comunicação) com a arquitetura de forma abstrata mediante a visibilidade dos serviços específicos de aplicações interativas publicados e oferecidos pela arquitetura.

## 3.5 Trabalhos Relacionados

Nesta seção serão apresentados os trabalhos relacionados à arquitetura *ispTV* que contribuíram para a proposta e o desenvolvimento. Serão destacadas as características de cada trabalho, apresentando suas contribuições acadêmicas e suas principais contribuições e diferenças para a arquitetura *ispTV*. Os trabalhos relacionados também têm como objetivo destacar diferentes implementações de arquiteturas de canal de interatividade para melhor apontar as características do trabalho proposto.

A seguir serão apresentados dois trabalhos relacionados, sendo o primeiro um protótipo de um canal de interatividade para a plataforma JiTV desenvolvido no padrão brasileiro de televisão digital (SBTVD). Propõe um PSI utilizando XMLs. O segundo propõe um modelo de canal de interatividade para o padrão americano (ATSC) utilizando uma arquitetura de *Web services*.

Ambos os trabalhos têm semelhanças com o apresentado neste trabalho, pois o primeiro trata de uma proposta para o SBTVD, e o segundo utiliza uma arquitetura baseada em *Web services*.

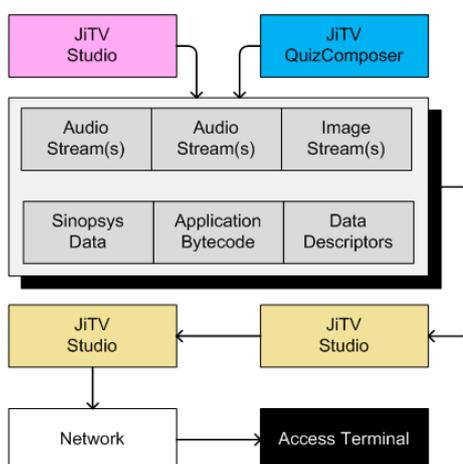
### 3.5.1 Canal de Retorno em Sistemas de Televisão Digital Interativa: Estratégias para a criação de aplicações utilizando um Provedor de Serviços Interativos

A plataforma JiTV, desenvolvida pelo TVDILab (Laboratório de Televisão Digital Interativa) residente na PUC Minas Campus Poços de Caldas é uma alternativa ao SBTVD.

Os programas interativos na plataforma JiTV (pTVDI) devem ser estruturados utilizando esquemas XML que descrevem as emissoras, programas e informações complementares (catálogo de gêneros, idiomas e close caption). Com as informações presentes nos arquivos XML, mídias e dados interativos, é possível adaptar o conteúdo dos programas interativos para os telespectadores (SANTOS JUNIOR *et al.*, 2007).

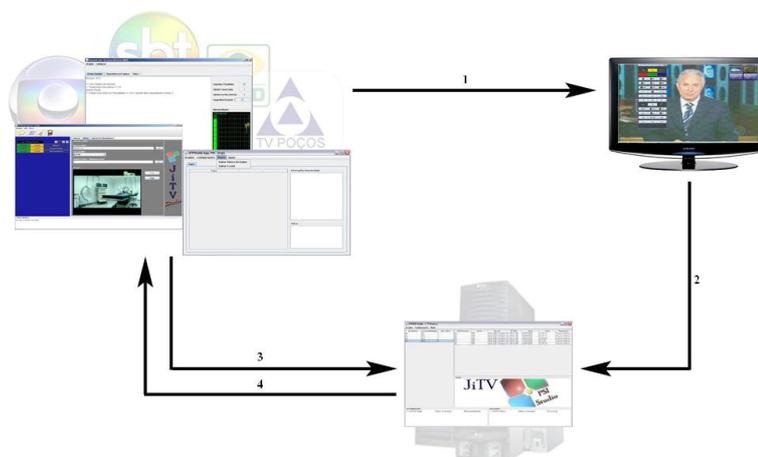
A plataforma JiTV oferece um conjunto de funcionalidades que suportam todos os ciclos de um programa interativo desde a criação e autoria até a recepção pelos telespectadores, oferecendo várias ferramentas com a finalidade de autoria e transmissão dos programas e garantindo a capacidade de visualização dos programas independente do terminal de acesso, como tv, pdas, computadores pessoais e celulares, dentre outros.

A figura 14 apresenta a arquitetura da plataforma JiTV.



**Figura 14:** Arquitetura Plataforma JiTV - adaptada de (SANTOS JUNIOR *et al.*, 2007)

Uma vez que a plataforma possui a capacidade de autoria dos programas, transmissão e recepção, os programas devem possuir suporte à comunicação com o meio externo para que os dados resultantes da interação sejam armazenados por um PSI. A plataforma JiTV também oferece um conjunto de ferramentas ao telespectador, canal de interatividade e emissora para o suporte as aplicações interativas que dependem do canal de interatividade como ilustrado na figura 15 (SANTOS JUNIOR. *et al*, 2009).



**Figura 15:** Estrutura PSI com plataforma JiTV

No telespectador, representado na figura 15 pelo fluxo 2, as aplicações interativas oferecem a característica de comunicação com o meio externo através do envio de arquivos XML, contendo o resultado da interação com o telespectador. O arquivo é enviado para o PSI para ser armazenado e futuramente utilizado para gerar informações para a emissora.

Na emissora, a aplicação JiTVStudio PSI possui a característica de transformar a necessidade de informação da emissora em dados para serem oferecidos pelo JiTVPSI Studio. Uma vez que esses dados estão prontos eles são enviados para o JiTVPSI Studio, representados na figura 15 pelo fluxo 3 e 4, e uma resposta é esperada para que as informações solicitadas sejam disponibilizadas para serem utilizadas pela emissora.

Na parte referente ao canal de retorno, existe um módulo intermediário entre emissora e telespectador com a capacidade de oferecer as funcionalidades de um PSI. Este módulo - JiTVPSI Studio - possui características de armazenamento e processamento dos dados. A característica de armazenamento é responsável por receber conexões dos telespectadores, aguardar o envio dos arquivos XML referentes à interação em execução e armazená-los. Já a funcionalidade de processamento de dados é responsável por aguardar conexões das emissoras solicitando informações referentes aos dados (XML) armazenados. Assim, permitindo que dados sejam coletados e informações sejam enviadas para a emissora quando solicitado.

Com estes componentes, a plataforma JiTV é capaz de contemplar os presentes na interatividade de um programa interativo, desde sua transmissão (fluxo 1), comunicação das aplicações interativas com PSI (fluxo 2), até a comunicação da emissora com o PSI (fluxo 3).

A plataforma JiTV relaciona-se à arquitetura proposta neste trabalho pelo oferecimento de suporte, tanto para a emissora quanto para o usuário, em relação às aplicações interativas. Uma vez que na plataforma JiTV todo o gerenciamento dos dados das aplicações interativas é feita em cima de arquivos XML, na arquitetura *ispTV* são dados processados por *Web services* e armazenados em um banco de dados. Referindo a forma de suporte à emissora, ambas trabalham semelhantemente, uma vez que a plataforma JiTV permite manipular os dados presentes no PSI e a arquitetura *ispTV* também. Já na arquitetura *ispTV*, é agregado APIs que facilitam o desenvolvimento de serviços e de aplicações interativas oferecendo funcionalidades na comunicação e publicação de serviços.

### **3.5.2 WRCS – Projeto de um Sistema de Canal de Retorno baseado na Web em broadcast de serviços de dados**

Para que o sistema de televisão digital americano ATSC-ACAP garanta o suporte a aplicações interativas one-way e two-tay um modelo de canal de interatividade deve ser utilizado. O *Web-based Return channel* (WRCS) sugere uma arquitetura de canal de interatividade baseada na web para oferecer suporte à grande variedade de interações que podem ser transmitidas via rede de broadcast e, conseqüentemente, a capacidade de realimentação do conteúdo através dos dados resultantes da interação dos telespectadores com os programas interativos (SEOK. *et al*, 2005). Outra importante característica é a capacidade de conexão de vários telespectadores.

Para garantir estes serviços sua arquitetura está dividida em três tipos de servidores e um switch com grande capacidade de distribuição (*layer 4 switch*). Os servidores são classificados em servidores de armazenamento (*database servers*), servidores de interatividade do canal de interatividade (*interactive channel servers*) e em servidores *Web services*.

Os servidores de armazenamento trabalham para armazenar as informações relacionadas a serviços não interativos e interativos transmitidos pela rede de broadcast. Com estas informações, é possível identificar serviços de *broadcast*, localização e tipo de outros servidores que estão relacionados aos dados de transmissão, aplicações interativas, canal de retorno, certificados dos usuários, contas de usuários, erros de sistema e lista de backup, dentre outros.

Os servidores de interatividade são representados como o cérebro dos serviços interativos do WRCS. Primeiramente esses servidores recebem dados relacionados ao destino da interatividade e a executam como programado. Outra característica deles é a transmissão das informações correspondentes às requisições dos usuários para os Web services.

Os *Web services* possuem a característica de gerenciar o controle de sessão para oferecer o suporte a conexões HTTP e TCP/IP feitas pelos terminais (STP) presentes nos telespectadores.

Para que os serviços possam ser manipulados de forma consistente o WRCS utiliza *Enterprise Java Beans* (EJB), que ajuda na simplificação e desempenho do desenvolvimento das aplicações interativas.

A validação do WRCS é feita utilizando uma aplicação interativa que disponibiliza uma lista de atores presentes em um programa televisivo. As informações da interação são enviadas para o WRCS através da infra-estrutura de rede disponível. Ao ser recebida, a certificação do usuário e sua validade são checadas. Sendo confirmadas, o resultado da interação é armazenado pelo WRCS e retornado para os usuários através da rede de broadcast ou por uma rede IP. O gerenciamento do WRCS pode ser feito através de uma aplicação web que possibilita controlar suas principais funcionalidades.

A justificativa deste trabalho relacionado à utilização de tecnologias web para a implementação deste modelo de canal de retorno (WRCS) é que o usuário inicie uma conexão, da mesma forma que o funcionamento das aplicações web (SEOK. *et al*, 2005).

As duas plataformas descritas possuem características que foram consideradas na proposição da arquitetura *ispTV* apresentada neste trabalho.

Ambos oferecem alternativas para a implementação de um canal de interatividade utilizando diversas tecnologias e padrões. O suporte multiusuário, a comunicação utilizando os protocolos TCP/IP e a utilização de um PSI são as semelhanças presentes entre os trabalhos relacionados e a arquitetura proposta.

Ambos os trabalhos utilizam a comunicação TCP/IP diferenciando-se em como os dados trafegam entre emissora, PSI e telespectador. Na plataforma JiTV, são utilizados arquivos XML para transporte dos dados e no WRCS é utilizada a tecnologia EJB para que as informações sejam transmitidas.

A arquitetura de canal de interatividade de ambos, diferenciando-se da forma de comunicação, são bem diferentes. Na plataforma JiTV uma aplicação Java gerencia toda a comunicação e acesso aos dados entre emissora e telespectador através de conexões multiusuário. Já no WRCS utiliza-se uma arquitetura de servidores e Web services para gerenciar as conexões.

A arquitetura *ispTV* proposta possuirá algumas das características presentes na plataforma JiTV e no WRCS, sendo organizadas de forma diferente para garantir todas as características propostas pela norma brasileira de televisão digital e pela especificação do middleware brasileiro Ginga. Essa organização também deve levar em consideração as delimitações físicas e a disponibilidade tecnológica brasileira.

Na arquitetura *ispTV* as ferramentas de suporte ao PSI são oferecidas baseadas na comunicação utilizando *Web services*, o que diferencia da forma utilizada pela plataforma JiTV. A organização das ferramentas presentes na emissora e no usuário são semelhantes, uma vez que a arquitetura *ispTV* oferece uma ferramenta de gerenciamento de serviços e suporte ao desenvolvimento de aplicações com comunicação com o canal de interatividade. Já as características baseada nos serviços, a arquitetura *ispTV* utiliza *Web services* como o modelo WRCS, diferenciando-se pela quantidade de camadas de servidores que possui o modelo e não está presente na arquitetura *ispTV*. Com isso, pode-se observar que a arquitetura *ispTV* possui algumas características presentes em ambos os trabalhos relacionados, mas não sendo implementados ou utilizados da mesma forma, e sim, com a intenção de simplificar o processo de utilização das aplicações interativas e seus respectivos serviços interativos.

As características que mais se destacam na arquitetura *ispTV*, podendo salientar como novos conceitos e funcionalidades para sistemas de TVDI, são: a forma de comunicação genérica oferecida pelas APIs, destacando as implementações Java e LUA; as formas de publicação de diferentes serviços em diferentes tecnologias; e a utilização de uma implementação serviço (serviço *ispTV*) próprio sem a utilização de *frameworks* para *Web services*, o que pode-se ser destacado junto as APIs de comunicação as contribuições e inovações diante os trabalhos relacionados.

Utilizando a arquitetura desenvolvida e discutida neste capítulo, serão descritas no próximo capítulo duas provas de conceito envolvendo o desenvolvimento de duas aplicações interativas, com seus respectivos serviços interativos, utilizando as funcionalidades oferecidas pela arquitetura *ispTV*.

# Capítulo 4

## APLICAÇÕES INTERATIVAS UTILIZANDO ISPTV

---

*Este capítulo apresenta aplicações interativas desenvolvidas no contexto das classes de aplicações interativas em conformidade com os objetivos descritos no decreto Brasileiro n°4.901.*

O capítulo anterior apresentou a arquitetura genérica de um PSI para sistemas de televisão digital, a arquitetura *ispTV*, e para contribuir com o processo de validação e teste da arquitetura, este capítulo apresenta aplicações interativas dentro do contexto brasileiro de Televisão Digital.

Neste capítulo serão apresentadas aplicações interativas com foco em aplicações educacionais (*t-learning*) com o intuito de prover melhorias para sistemas de ensino a distância e de oferecer suporte aos telespectadores com necessidades especiais. Outra classe de aplicação abordada é chamada de aplicações de realidade virtual baseadas em computação em nuvem que podem utilizar as aplicações interativas para interação em ambientes de realidade aumentada.

### 4.1 Aplicações Educacionais

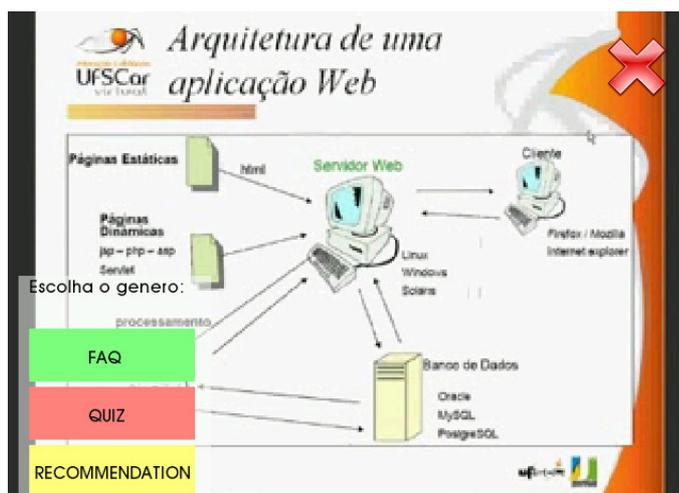
As aplicações educacionais desenvolvidas para utilização neste trabalho partiram do princípio de demonstrar algumas das formas de inclusão social que podem ser utilizadas pelas emissoras de televisão ao desenvolverem suas aplicações interativas que estarão presente nos programas televisivos. Para o desenvolvimento da aplicação interativa foi utilizado o *middleware* brasileiro na implementação Ginga-NCL, oferecendo o suporte a aplicações NCL em conjunto

com sua linguagem de *script* Lua, e a API de comunicação *ispTV Client API* para envio e recebimento das informações resultantes da interação do telespectador/usuário com as aplicações interativas.

As aplicações desenvolvidas estão divididas em quatro principais aplicações de suporte ao usuário: a aplicação interativa geral gera três outras aplicações, de modo que possa ser transmitida através de vários programas e que apenas o seu conteúdo seja alterado. As outras aplicações interativas que podem ser acessadas pela aplicação interativa geral são referentes a disponibilidade de FAQs (Frequently Asked Questions), quiz e recomendação de programas baseados no conhecimento, relacionado ao conteúdo transmitido, do usuário.

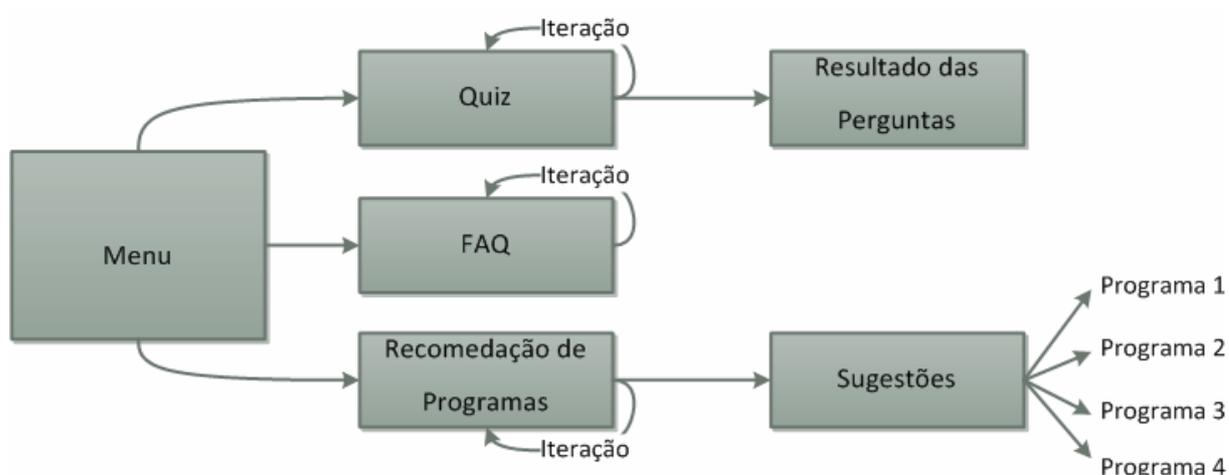
Para tornar possível a apresentação desses aplicativos para usuários, as emissoras de televisão devem-se concentrar na produção de conteúdo televisivo que desejam oferecer aos usuários. Isso porque as aplicações interativas desenvolvidas dependem e estão interligadas ao conteúdo produzido para que os assuntos abordados nas aplicações estejam sincronizados com os dos programas.

As aplicações demonstram uma relação com o programa interativo, pois são uma referência para a implementação das três aplicações para os educadores e os usuários. A aplicação interativa geral transmitida junto ao programa interativo possui um menu que disponibiliza o acesso as outras três aplicações interativas. Cada aplicação interativa específica tem um recurso oferecido de apoio educacional, o qual pode ser utilizado tanto pelos educadores e pelos próprios usuários, que são os alunos ou pessoas interessadas no assunto transmitido.



**Figura 16:** Aplicação Interativa de Educação com menus para aplicações específicas

A aplicação específica de FAQ tem como objetivo oferecer aos telespectadores/usuários um conjunto de perguntas feitas frequentemente com relação ao assunto que está sendo transmitido. Com a aplicação, permite-se que o programa interativo disponibilize as questões mais acessadas e que os usuários tenham acesso às perguntas e respostas respondidas pelos produtores de conteúdo. Outra aplicação disponível é a de quiz, no qual as perguntas são disponibilizadas aos usuários para serem respondidas. Esta aplicação pode ser utilizada pelos produtores de conteúdo para questionarem ou obter informações dos usuários sobre o programa transmitido e é utilizada como base para a aplicação de recomendação. A aplicação de recomendação de programas obtém as informações referentes às respostas do quiz e informa o usuário se ele está no programa televisivo que mais se assemelha as suas respostas, recomendando que o usuário troque de canal para acompanhar a programação que possui mais conhecimento. No fluxograma apresentado pela figura 16, é possível visualizar as possibilidades de interação do usuário descritas anteriormente.



**Figura 17:** Fluxograma de interação

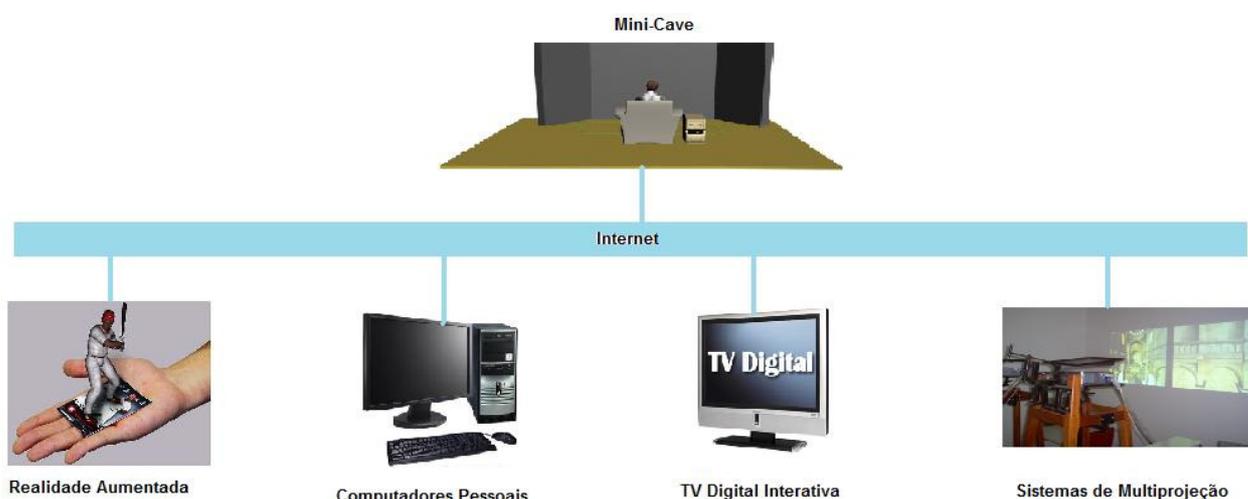
A comunicação com a arquitetura *ispTV*, em que as perguntas e respostas são enviadas no sentido aplicação/*ispTV* e *ispTV*/aplicação, foi feita utilizando a *ispTV Client API* do lado das aplicações. Na arquitetura *ispTV*, foi criado um serviço específico para cada tipo de aplicação específica, em que as perguntas eram previamente cadastradas em um banco de dados *SQL* e assim que fossem solicitadas, eram oferecidas aos alunos. Para as aplicações interativas que utilizavam o formato de quiz as respostas eram armazenadas em um outro serviço específico que oferecia o resultado final em forma de *score* ou em uma recomendação de que a programação que estava sendo visualizada, era a mais indicada para o conteúdo respondido.

## 4.2 Aplicações de Realidade Virtual

Outra aplicabilidade da arquitetura apresentada neste trabalho foi destacada na utilização de outras plataformas distribuídas de transmissão de mídia de realidade aumentada (PRADO, G.M.; et AL., 2011). A sua utilização está inserida no projeto XPTA (XPTA.LAB) de uma plataforma para virtualidade imersiva e interativa baseada em computação em nuvem, ou seja, o desenvolvimento de uma estrutura básica de comunicação em nuvem, com o apoio e desenvolvimento de conteúdos de

realidade virtual multiusuário distribuída e, também, para TV digital, dispositivos móveis (smartphones, PDAs) e computadores pessoais.

No entanto, ainda é difícil hoje em dia desenvolver a realidade virtual e aumentada em aplicações colaborativas da Web aberta, devido à dificuldade em sincronizar os usuários e transferir grande quantidade de dados em tempo real. A infraestrutura de Internet atual, entretanto, permite que esses aplicativos sejam desenvolvidos e os desenvolvedores obtenham vantagens com uma plataforma (ou quadro) que abstraia os aspectos complexos da computação imersiva, permitindo o foco nas principais funcionalidades de suas aplicações.



**Figura 18:** Aplicações Imersivas e Interativas utilizando *ispTV*

A Figura 18 mostra a visão geral do projeto, cuja as aplicações de televisão digital interativa estão se interligando com aplicações imersivas e de realidade aumentada através de uma Mini-cave, que é responsável por gerenciar e sincronizar todas as aplicações e possibilitar a interação com o usuário. A arquitetura *ispTV* é integrada a Mini-cave com a inclusão da arquitetura no cluster de gerenciamento de aplicações. Com isso, aplicações interativas imersivas e de realidade virtual podem ser executadas no ambiente.

O propósito geral da utilização da arquitetura *ispTV* no projeto XPTA é possibilitar que aplicações executadas na Mini-cave, possam oferecer formas interativas de visualização para os telespectadores dos programas interativos e que possam interagir diante de aplicações de realidade aumentada.

As aplicações interativas que serão utilizadas junto ao projeto XPTA são de manipulação de mídias pelos usuários através da televisão. As aplicações que estão dentro deste contexto geralmente estão relacionadas à navegação em ambientes 3D que são atualizados em tempo real por uma aplicação interativa como controle de navegação. Um exemplo seria o controle de jogos através da televisão, em que uma aplicação interativa assume o controle do personagem em questão e envia a mini-cave as coordenadas que devem ser executados para a atualização da direção do personagem na exibição da mídia. O funcionamento deste tipo de aplicação é baseado no envio das mídias televisivas para os usuários que estão sendo transmitidas pela mini-cave junto a uma aplicação de controle da mídia. O usuário pode navegar passando direções e destinos, através da aplicação interativa, enviando estas direções para a mini-cave que atualiza a mídia a ser gerada com relação aos movimentos enviados pelo usuário. Outra aplicabilidade seria a navegação em obras de arte em 3D, em que o usuário tem o controle de navegação em uma obra e pode navegar dentro dela através das mídias transmitidas e a aplicação interativa de controle de mídia transmitida.

### **4.3 Aplicações Interativas para Inclusão de pessoas com necessidades especiais**

Outro estudo de caso desenvolvido para a utilização da arquitetura proposta está de acordo com o decreto brasileiro n 4.901, cujo o objetivo é possibilitar o máximo de acesso e inclusão social através das aplicações interativas. Com isso, uma aplicação interativa para pessoas com necessidades especiais relacionadas à visão foi desenvolvida com o objetivo de comprovar que é possível abranger todos os tipos de aplicações para que os objetivos descritos no decreto brasileiro sejam alcançados.



**Figura 19:** Aplicação Interativa XPTA

Na figura 19 é demonstrada a aplicação que tem seu conteúdo relacionado a um monumento histórico na Paraíba (Paraíba - Brasil), que é representado por uma pedra chamada Pedra do Ingá (Pedra do Ingá, 2011). Devido a isso, muitas perguntas que variaram entre questões históricas e curiosidades estão disponíveis para os telespectadores.

Como ilustrado na figura 19, o programa interativo é realizado juntamente com transmissão de programa brasileiros, disponibilizando ao usuário um menu com as opções de história e curiosidade sobre o patrimônio histórico. Ao mesmo tempo, um áudio narrando todo o conteúdo do texto na tela é iniciado. Depois de selecionar alguma das opções, serão oferecidos ao usuário três questões relacionadas ao gênero escolhido, no qual o usuário terá a opção de escolher entre três alternativas. Ao responder uma das três perguntas, uma tela final será exibida com o resultado final da interação. O resultado tem informações sobre quais as perguntas urge lembrar o usuário respondeu corretamente ou não e sua pontuação final. Lembrando que todo o processo de interação é seguido de uma narração.

A aplicação interativa para pessoas com necessidades especiais visuais também utiliza as funcionalidades oferecidas pela arquitetura *ispTV* onde as perguntas e respostas são enviadas no sentido aplicação/*ispTV* e *ispTV*/aplicação utilizou-se a API *ispTV C*. Com isso, foi criado um serviço específico para a aplicação específica onde as perguntas foram cadastradas em um banco de dados *SQL* e oferecidas assim que solicitadas pela aplicação interagida pelos alunos.

As aplicações interativas desenvolvidas como prova de conceito da arquitetura *ispTV* são todas de caráter informativo e com foco educacional, onde pode-se incluí-las em um dos objetivos de inclusão social presente no decreto brasileiro. Com isso, seu funcionamento utiliza as funcionalidades oferecidas pela arquitetura *ispTV* agregando as características de comunicação com o meio externo através de um canal de interatividade. As aplicações interativas utilizam todas as funcionalidades oferecidas pela arquitetura *ispTV* para prover a comunicação com o meio externo através da utilização da API Cliente *ispTV C*, para oferecer simplicidade na comunicação, e a simplicidade de desenvolvimento de serviços locais – serviços *ispTV* – utilizando apenas classes Java publicando-as na arquitetura para serem acessadas pelas aplicações.

Com o desenvolvimento das aplicações, consegue-se fazer provas de conceito das funcionalidades desenvolvidas para a arquitetura *ispTV* que utilizem a comunicação externa com canal de interatividade e o armazenamento e oferecimento de informações a partir de serviços específicos de aplicação presentes na arquitetura *ispTV* como um Provedor de Serviços Interativos.

# Capítulo 5

## VALIDAÇÃO DE USABILIDADE E DESEMPENHO

---

*Este capítulo apresenta validações de usabilidade das APIs de comunicação com a arquitetura ispTV e testes de desempenho de comunicação com a arquitetura.*

Para analisar a arquitetura do ponto de vista de usabilidade, foram elaborados e aplicados testes no contexto de utilização das APIs oferecidas pela arquitetura para simplificar o desenvolvimento da comunicação através do canal de interatividade, por parte dos desenvolvedores. Referente ao aspecto arquitetural criaram-se e aplicaram-se testes que mediram capacidade e escalabilidade de comunicação entre aplicações e a arquitetura, medindo o tempo de resposta e capacidade de processamento de requisições.

O teste de usabilidade apresenta as implementações Java e LUA da API cliente *ispTV* de forma a demonstrar com é feita a sua utilização no desenvolvimento de aplicações interativas. O ponto de destaque deste teste é demonstrar experiências na utilização da API pelos desenvolvedores demonstrando os pontos significativos relatados da API de comunicação.

Para verificar o real comportamento do sistema em termos de tempo e capacidade de processamento, foram realizados dois tipos de testes. O primeiro deles são testes envolvendo o tempo de resposta de uma requisição feita por uma aplicação Java de *benchmark* para o *ispTV* para identificar quanto tempo leva-se para fazer certos tipos de requisições a arquitetura. O segundo é um teste de escalabilidade cujo o foco é demonstrar quando a arquitetura começa a perder desempenho diante as requisições.

Com esses testes pode-se ter um entendimento maior de quanto tempo levaria para a aplicação interativa, transmitida pela emissora, fazer uma requisição de um

informação e quão escalável é a arquitetura. Com os resultados obtidos serão feitas análises e podem-se propor novas estratégias de melhoria para o desempenho da arquitetura caso necessário. A seguir serão descritos os testes executados e os resultados obtidos, com detalhamento por tabelas e gráficos para o melhor entendimento.

Uma vez que a arquitetura proposta neste trabalho é responsável por oferecer estes tipos de serviços, o tempo de resposta das requisições feitas para a arquitetura *ispTV*, inserida no cenário do SBTVD, foi analisado para oferecer informações mais precisas de desempenho. Outro fator a ser analisado na arquitetura diante o cenário global do SBTVD é a capacidade de oferecer suporte a todas as requisições dos programas interativos que são feitas a um serviço específico de aplicações interativas. Por exemplo, em um processo de interação com um programa televisivo que ofereça uma aplicação interativa e se o serviço específico é capaz de suportar N requisições vindas das aplicações interativas sendo acessadas de lugares diferentes. Com isso, a seguir serão demonstrados testes, tempo de resposta e escalabilidade, para validar a arquitetura *ispTV* demonstrando sua capacidade relacionada a execução de diversas aplicações interativas em um mesmo momento.

## 5.1 Teste de usabilidade

O teste de usabilidade é baseado na utilização das implementações Java e LUA da API de comunicação. As implementações podem ser utilizadas tanto em aplicações interativas desenvolvidas em Java quanto em NCL. No caso de Java, as aplicações devem utilizar a implementação Java e, no caso de aplicações interativas desenvolvidas em NCL, deve-se utilizar a implementação LUA, pelo fato de LUA ser a linguagem de *script* utilizada pelo NCL. As formas de utilização destas implementações serão demonstradas na seção 5.1.1.

O teste, utilizando a implementação Java, será feito pela equipe de desenvolvimento da consultoria Breitech Soluções e Serviços residente em São

Paulo – SP onde atua em projetos financeiros para o Banco Votorantim e apresentará as considerações e relatos dos desenvolvedores diante a utilização da API para o desenvolvimento da comunicação com serviços interativos oferecidos pela arquitetura *ispTV*. Já o teste utilizando a implementação NCL será feito pelos desenvolvedores da empresa Mvar Soluções e Serviços com o intuito de demonstrar o processo de entendimento de desenvolvedores Java na utilização da API LUA no desenvolvimento da comunicação para as aplicações interativas NCL.

### **5.1.1 Implementações da API *ispTV***

A utilização da API Java pode ser feita através de simples configurações de uma requisição, em que cada requisição possui um endereço final, nome do serviço interativo a ser acessado e a operação a ser executada pelo serviço. Diante da configuração, a requisição é feita passando os parâmetros de consulta para o serviço e a mensagem de resposta é retornada. A figura 20 demonstra a utilização da API em sua total forma de utilização:

```
1 package isptv.client.test;
2
3 import isptv.client.bean.Message;
4 import isptv.client.core.Request;
5
6 public class TestRequest {
7
8
9     public static void main(String [] args) {
10
11         String ispTVEndPoint = "http://breitech.no-ip.info:8080/ispTV/services/broker";
12
13         Request requisicao1 = new Request(ispTVEndPoint, "IspTVService1", "echo", null);
14
15         /*
16          * Cria string de envio, podendo ser um conjunto de strings.
17          * String pode ser:
18          * . XML
19          * . Qualquer tipo de primitivo (Wrapper)
20          */
21         String conteudoEnvio = "xml or other things";
22         Message mensagemResposta = requisicao1.send(conteudoEnvio);
23
24         // Aplica utilizacao da resposta.
25         if (mensagemResposta != null) {
26             String resposta = (String) mensagemResposta.getResult();
27             System.out.println("Resposta: " + resposta);
28         } else {
29             System.out.println("Nenhum dado retornado.");
30         }
31     }
32 }
33
34 }
35 }
```

Figura 20: Utilização implementação Java API comunicação *ispTV*

A utilização da implementação Java se resume em três passos: definição do *endPoint* da arquitetura *ispTV*, envio da requisição e obtenção da resposta final da requisição. Na linha 11 é feita a definição do *endPoint* em que a arquitetura *ispTV* está presente; na linha 13 é montada a requisição sendo atribuído o *endPoint*, o serviço interativo e sua operação que será acessada, respectivamente; na linha 21 é montado o conteúdo a ser enviado na requisição, podendo ser formado por primitivos e um xml contendo as informações a serem enviadas. O número de conteúdo a ser enviado é variável por serviço interativo, dependendo da regra de negócio; Se o serviço obter alguma resposta, na linha 26 é obtida a informação processada e disponível na mensagem de resposta do serviço interativo. Com isso, pode-se utilizar a API para enviar dados aos serviços e obter informações para serem utilizadas nas aplicações interativas.

A utilização da API LUA possui as mesmas lógicas para envio da mensagem, e pode ser feita através de simples configurações de uma requisição, em que cada requisição possui um endereço final, nome do serviço interativo a ser acessado e a operação a ser executada pelo serviço. A figura 21 demonstra a utilização da API em sua total forma de utilização:

```
1 -- Configura o endpoint da arquitetura ispTV.
2 setEndPoint("http://breitech.no-ip.info:8080/ispTV/services/broker")
3
4 -- Informações a serem enviadas.
5 local params = {}
6 params[0] = "Informação 1"
7 params[1] = "Informação 2"
8 params[2] = "Informação 3"
9
10 -- Envia a mensagem para o serviço específico com a operação a ser utilizada.
11 sendMessage("IspTVService1", "echo", params)
12
13 -- Obtem a resposta que é preenchida quando o callback da requisição termina.
14 local resposta = getResponse()
15
16 -- 100 100 são o x e o y no pixel que a mensagem será exibida.
17 canvas:drawText(100, 100, "Resposta: "..resposta)
18
```

**Figura 21:** Utilização implementação LUA API comunicação *ispTV*

A utilização da implementação LUA se resume nos mesmos passos que a implementação Java para que seja seguido o mesmo padrão de configuração de envio de requisições. Na linha 2 é feita a definição do *endPoint* em que a arquitetura *ispTV* está presente; na linha 5 é montado o conteúdo a ser enviado na requisição, podendo ser formado por primitivos e um xml contendo as informações a serem enviadas; na linha 11 é montada a requisição, sendo utilizado o *endPoint* já atribuído na linha 2, o serviço interativo e sua operação que será acessada, respectivamente; na linha 14 é obtida a informação processada e disponível na mensagem de resposta do serviço interativo, sendo impressa na tela na linha 17.

Com a descrição da utilização de ambas as APIs oferecidas pela arquitetura, na seção a seguir serão apresentadas as considerações de usabilidade perante as duas APIs.

### 5.1.2 Teste de usabilidade da API *ispTV*

O teste de usabilidade elaborado, a ser aplicado na arquitetura *ispTV*, foi submetido a um grupo composto de 6 pessoas, desenvolvedores de software, nos níveis Junior, Pleno e Sênior. A distinção de níveis foi escolhida para obter um *feedback* mais preciso quanto ao nível técnico dos desenvolvedores de *software* perante a API.

O teste de usabilidade tem como objetivo obter o *feedback* diante três quesitos: abstração da implementação, simplicidade de utilização e simplicidade de manipulação da informação.

Na abstração da implementação será analisado se a API pode ou não ser mais abstrata do que ela já é, simplificando mais a vida do desenvolvedor e abstraindo detalhes de comunicação. A simplicidade de utilização será analisada a qualidade da abstração da API. E a simplicidade de manipulação de informação analisará se o desenvolvedor tem flexibilidade de envio de informações e recebimento de informações de forma simples.

Para todos os níveis de desenvolvedores foi apresentada a API de acesso a arquitetura *ispTV* e oferecido um endereço onde a arquitetura esta localizada (<http://breitech.no-ip.info:8080/ispTV>). Com isso, o desenvolvedor consegue cadastrar novos serviços, ou utilizadas os já existentes, para poder testar a API e fazer sugestões de usabilidade.

Para a análise envolvendo os desenvolvedores Junior, as considerações e contribuições obtidas estiveram relacionadas a facilidade de enviar e receber mensagens utilizando um serviço web *ispTV*. Em respeito ao envio e recebimento de mensagem, foi questionado o que era utilizado e se existiam formas melhores de simplificar o objeto de retorno limitando a apenas informações retornadas pelo serviço. O que serve como contribuição, pois a mensagem que é retornada utiliza as mesmas características da mensagem enviada, contendo os mesmos atributos, e diferenciando pela forma de obtenção da informação.

Para a análise envolvendo os desenvolvedores Pleno, as considerações foram mais objetivas diante o objetivo da API e o que ela se propõe a fazer. Com isso, os

desenvolvedores conseguiram utilizar a API da forma que foi desenvolvida dizendo que o propósito da API era enviar o máximo de informação possível na mensagem e receber o máximo de informação do retorno do serviço. Com a utilização do mesmo padrão de mensagem tanto para envio quanto para recebimento, este objetivo foi alcançado. Um ponto destacado pelos desenvolvedores Pleno, foi a capacidade de reutilizar a mesma requisição para diferentes mensagens, o que torna a reutilização de código desenvolvido menor e simplifica o desenvolvimento de programas que utilizem a API.

Já a análise feita pelos desenvolvedores Seniores foi mais criteriosa diante a implementação e os objetivos a serem atingidos com a API. Para deixar mais genérica e proporcionar a maior reutilização de código, foi sugerido que uma requisição fosse configurada com um endereço final de conexão (endPoint) e com posse desta requisição, as mensagens fossem disparadas sem a necessidade de conhecimento do endPoint. Para isso, sugeriu-se a criação de um construtor que receba o endPoint e configure a conexão com a arquitetura *ispTV*. Com a configuração feita, o processo de envio que é utilizado se mantém o mesmo sem a necessidade de especificar para onde a requisição está sendo enviada. Mesmo com sua criticidade na análise, foi destacado que a API está fazendo o que se propõe de forma coesa e objetiva.

Com as análises feitas, podemos concluir que a API oferece formas claras de utilização e simplificação de envio de mensagem para a arquitetura *ispTV* abstraindo configuração de comunicação do desenvolvedor sem restringir qualquer funcionalidade de envio e obtenção de informações através da arquitetura.

## 5.2 Teste de tempo de resposta

Os testes envolvendo tempo de resposta foram planejados para englobar a comunicação entre aplicações Java de *benchmark* utilizando serviços sem a intervenção da arquitetura *ispTV*. Isto para ocorrerem comparações de como a

arquitetura proposta e desenvolvida acarreta na comunicação com os serviços publicados.

O primeiro passo para a execução dos testes foi desenvolver três tipos de serviços a serem utilizados separadamente e/ou junto com a arquitetura. Foi desenvolvido um serviço para oferecer apenas o recebimento de uma informação, simulando aplicações interativas que apenas precisem enviar um requisição e receber uma resposta simples e imediata, sendo classificado em teste de *delay* e com o intuito de demonstrar apenas o tempo de comunicação ou resposta da aplicação interativa com o serviço específico. Também foram desenvolvidos dois outros tipos de serviços que oferecem, de forma simples, funcionalidade de processamento matemático, simulando aplicações interativas que precisem de processamento complexo, sendo classificado em teste de processamento e serviço de conexão com um banco de dados, simulando aplicações que possuem informações a serem manipuladas em um banco de dados, sendo classificado em teste de banco de dados. Estes dois últimos são serviços que demonstram como a arquitetura se comporta com este tipo de funcionalidade presentes nos interesses das aplicações desenvolvidas.

O ambiente de teste montado para o teste de tempo de resposta é baseado em um servidor de serviços específicos de aplicações interativas contendo os serviços mencionados anteriormente e uma aplicação *benchmark* desenvolvida em Java para gerar as requisições para cada serviço específico. O ambiente é composto por um computador que possui o servidor específico e um outro computador possuindo a aplicação Java de *benchmark*. O número de requisições feitas para cada tipo de serviço desenvolvido, demonstrado nas tabelas a seguir, foram de 300. Isto para que uma média de tempo e uma precisão maior fosse obtida com as requisições.

Ambos os tipos de testes executados possuem três análises, baseada em tempo por ms (milissegundos), a serem feitas. A primeira é a medição feita sem a utilização da arquitetura *ispTV* (Sem *ispTV*), sendo processada através da comunicação de uma aplicação Java *benchmark* diretamente com o serviço desenvolvido; a segunda é feita utilizando a funcionalidade de publicação de serviços oferecida pela arquitetura *ispTV* só que com o cadastro de serviços locais

(ispTV Local), ou seja, serviços que estão localizados no mesmo local da arquitetura; e por fim, a terceira medição é feita utilizando a arquitetura *ispTV* e a publicação de serviços que estão em outras localidades (serviços JAX-WS e Axis2), ou seja, apenas é feito o cadastro do serviço e eles estão localizados em outro PSI.

**Tabela 1:** Teste baseado apenas em delay (Tempo de Resposta)

<b>Delay em <i>ms</i></b>	<b>Sem ispTV</b>	<b>ispTV Local</b>	<b>ispTV Remoto</b>
<b>ispTV</b>	-	512	-
<b>JAX-WS</b>	407	617	1.109
<b>Axis2</b>	412	642	1.121

Na tabela 1, são demonstrados os tempos resultantes da medição envolvendo apenas o tempo de resposta de uma requisição. Na primeira coluna é demonstrado uma certa simetria entre os tempos envolvendo as tecnologias JAX-WS e Axis2, que está em torno de 400 ms. O valor 0 é atribuído ao tipo de serviço *ispTV*, pois este não utiliza comunicação via rede para ser acessado, e sim, comunicação entre classes java. Analisando a segunda coluna, referente aos serviços publicados utilizando a arquitetura *ispTV*, já se tem tempo atribuído a utilização dos serviços *ispTV*. Isso mostra uma pequena diferença para os serviços que utilizam comunicação com a rede, uma vez que estes serviços não estão localizados em outros PSI, assim não possuindo comunicação via Internet com o meio externo. Já a terceira coluna, onde mostram que os serviços publicados em outro PSI apresentam uma diferença significativa no tempo de resposta para o serviço local da arquitetura (*ispTV*), chegando a ser o dobro do tempo de resposta.

**Tabela 2:** Teste baseado em processamento matemático

<b>Processamento em <i>ms</i></b>	<b>Sem ispTV</b>	<b>ispTV Local</b>	<b>ispTV Remoto</b>
<b>ispTV</b>	-	1765	-
<b>JAX-WS</b>	1441	2112	2437
<b>Axis2</b>	1502	2147	2462

Na tabela 2, o processo de medição é o mesmo, diferenciando-se apenas pelo serviço que é acessado. Neste caso o serviço não se limita apenas a responder a requisição, mas também a aplicar cálculos matemáticos (soma, subtração,

multiplicação e divisão) diversas vezes para simular um processamento simples. Podendo observar uma grande diferença no tempo de resposta das requisições.

**Tabela 3:** Teste baseado em acesso a Banco de Dados

<b>Banco de Dados em <i>ms</i></b>	<b>Sem ispTV</b>	<b>ispTV Local</b>	<b>ispTV Remoto</b>
<b>ispTV</b>	-	2216	-
<b>JAX-WS</b>	2070	2671	2902
<b>Axis2</b>	2127	2687	2913

Na tabela 3, tem-se o mesmo tipo de análise que na tabela 2, diferenciando apenas pelo serviço acessado e o que este serviço executa de processamento internamente. Neste caso, uma conexão com o Banco de Dados é aberta e uma solicitação simples de informação é feita.

**Tabela 4:** Teste baseado nas requisições da arquitetura

<b>Em <i>MS</i></b>	<b>ispTV</b>
<b>Delay</b>	752
<b>Processamento</b>	2.114
<b>Banco de Dados</b>	2.600

Na tabela 4, não é de interesse da medição qual tecnologia de publicação de serviços está sendo utilizada, mas sim qual é o comportamento da arquitetura, sendo transparente de tecnologia para todos os casos já apresentados nas tabelas anteriores. Os tempos apresentados são resultados de uma média dos tempos obtidos de requisições para cada tipo de tecnologia, levando em conta a publicação de diversos tipos de serviços em diversas tecnologias.

Analisando as quatro tabelas, pode-se chegar à conclusão que a tecnologia de serviço da arquitetura *ispTV* é mais rápida por não precisar utilizar a conexão com a Internet para redirecionamento de requisição. Já as tecnologias JAX-WS e Axis2 possuem um comportamento muito semelhante nas várias medições, o que trás um retorno interessante por não apresentar grandes diferenças de desempenho das tecnologias utilizadas. Por fim, quando se analisa o comportamento geral da arquitetura, sem se deter separadamente em cada tecnologia, seu desempenho em tempo de resposta é muito semelhante do que quando acessa-se separadamente as

tecnologias utilizadas. Isso se deve pelo rápido redirecionamento de requisições desenvolvido pela arquitetura *ispTV*. Uma vez que o tipo de serviço tende para o específico da arquitetura (*ispTV Service*), o tempo irá diminuir e por sua vez se tende para as tecnologias JAX-WS e Axis2 irá aumentar, isto pela simplicidade do serviço local *ispTV Service*.

Os testes realizados na seção anterior objetivam verificar o *delay*, tempo de resposta, resultante da comunicação de aplicações Java *benchmark* com a arquitetura *ispTV*. Eles demonstram o comportamento para poucas requisições de um serviço e podem apresentar variação diante um número elevado de requisições.

A seguir serão demonstrados testes de escalabilidade para verificar quantas requisições a arquitetura pode suportar oferecendo qualidade de resposta das informações.

### 5.3 Teste de escalabilidade

Através dos testes realizados seção anterior não se é possível concluir o quanto a arquitetura pode atender de requisições sem perca de desempenho e qualidade. Para tal análise, uma avaliação concreta da capacidade da arquitetura deve ser feita para analisar o número de requisições simultâneas e o tempo de resposta diante a sobrecarga de requisições. O teste a ser executado com a aplicação Java *benchmark* e o servidor com a arquitetura *ispTV* é baseado na capacidade que a arquitetura possui em atender o maior número de requisições em um tempo aceitável de requisição sem dar início a perca de requisições.

Para a execução dos testes, existia a necessidade de criação de uma aplicação em Java de *benchmark* com a capacidade de envio de requisições de forma a garantir um comportamento multitarefa e simular o acesso/requisição de diversas aplicações interativas aos serviços oferecidos. Esta aplicação tem como função enviar diversas requisições para a arquitetura com a finalidade de simular o acesso de diversos aplicações interativas simultaneamente na arquitetura. Para garantir esta funcionalidade, a aplicação foi desenvolvida utilizando *threads* que efetuam uma conexão com a arquitetura, seguidas de uma requisição e,

consequentemente, da recepção da resposta por parte da aplicação. O número de *threads* é variável por aplicação, podendo assim ser utilizadas em outros computadores e dispositivos para aumentar a carga de requisições. Todo esse processo de comunicação com a arquitetura leva um tempo para ser concluído e para armazenar este resultado para análise de escalabilidade, este tempo é salvo em um arquivo logo após o recebimento da informação. Na arquitetura, serviços foram desenvolvidos e publicados, dividindo-se em serviços *ispTV* locais e remotos, para simular uma situação bem próxima da real diante as requisições feitas pelas aplicações interativas.

As variáveis a serem utilizadas no cenário de teste de escalabilidade foram definidas para o número de aplicações interativas que serão executadas simultaneamente, para que seja definido o número de *threads* que a aplicação Java de *benchmark* possa simular as requisições simultâneas das aplicações interativas. Uma aplicação Java de *benchmark* foi estruturada para que possa executar 100 *threads* sem que elas comecem a sofrer interferência bruta do sistema operacional para poderem ser executadas, este número simula 100 requisições de aplicações interativas. Neste caso, foi analisada a possibilidade de utilização de cinquenta, setenta e cem *threads* por aplicação *benchmark*. Dentre os três valores a serem utilizados, setenta *threads* foi o valor escolhido por não exigir tanto do processamento extra do computador que está executando a aplicação *benchmark* para gerenciamento das *threads*. No caso de cinquenta, o sistema ficou muito ocioso, e no caso de cem, começou a apresentar sinais de sobrecarga. Com o número por aplicação *benchmark* definidos, o servidor que recebera as requisições foi definido com a instalação da arquitetura *ispTV* para receber as requisições.

O número de computadores que executarão as aplicações de *benchmark* foi definido por um pré-teste de quanto que a arquitetura começava a apresentar sobrecarga na resposta das requisições. Chegou-se ao número de oito computadores executando 70 *threads* com a aplicação *benchmark* instalada e foi definido o número de computadores para a execução do teste de escalabilidade. O número de computadores também foi escolhido, devido à intenção de resultado final deste teste, que é demonstrar quando a arquitetura *ispTV* começa a perder desempenho diante a rajada de requisições. Outra variável a ser analisada no

cenário são os servidores utilizados onde a arquitetura *ispTV* esta instalada. Para esta variável foram escolhidas três configurações de servidores descritos a seguir junto às outras variáveis utilizadas no teste:

**Tabela 5:** Cenários de teste

	<b>Nós</b>	<b>Requisições/Nó</b>	<b>Servidor</b>
<b>Cenário 1</b>	8	70	Pentium 4 - 1 Core - 1 GB Ram
<b>Cenário 2</b>	8	70	Core 2 Duo - 2,26 Ghz - 2 GB Ram
<b>Cenário 3</b>	8	70	Core 2 Duo - 2,26 Ghz - 4 GB Ram

Os cenários demonstrados na tabela 5 são semelhantes, analisando-se o lado da aplicação Java *benchmark*, em que sempre serão oito nós executando setenta requisições por cada aplicação *benchmark*, o que resulta em uma média de 560 requisições a serem feitas durante todo o processo de teste aos servidores. Já do lado dos servidores existe uma diferença referente às configurações com o objetivo de possuir formas de comparação de quando cada servidor começa a perder desempenho referente às requisições.

O processo de execução para os três cenários tem início com apenas um nó, que inicia o processo de requisição com as setenta *threads* e só para de enviá-las quando todo o processo do teste for concluído. Consecutivamente, até terminarem os nós, será iniciado o mesmo procedimento com todos até puder ser feitas observações relevantes que afetam a capacidade de funcionamento da arquitetura. Um fator importante é a ausência de interferência comunicação externa de rede de comunicação, executando os testes em um rede local e restrita para os oito nós e o servidor. Com todas as variáveis definidas, seguiram-se os testes da arquitetura.

A realização dos cenários de testes tem início com a execução de um nó por vez e pausadamente, o que possibilita uma análise diferenciada no instante em que cada nó começa a participar da execução de certo cenário. As análises dos cenários são feitas através dos logs gerados pelas aplicações geradas de requisição. Cada nó possui seu respectivo arquivo de log e cada requisição respondida pelo servidor possui uma linha correspondente no arquivo log, o que possibilita analisar o comportamento de cada nó em diferentes momentos. Isto permite a análise do

comportamento em relação à quantidade de nós que a arquitetura começa a apresentar sobrecarga.

Analisando o primeiro cenário, os oito nós trabalharam exaustivamente durante todo o processo da execução do teste. A cada novo minuto de execução do teste iniciava-se um nó novo, ou seja, cada nó que era incluído armazenava o log de  $n$  requisições com os resultados sendo afetados pela concorrência dos demais nós. Pode-se observar que ao iniciar o quarto e o quinto nó, a arquitetura começou a apresentar um *delay* mais significativo em relação ao início, até então, do terceiro nó. Este delay, que era no início de algo em torno dos 700/800 ms, passou a ser algo em torno de 2000/2300 ms. Com isso, para o primeiro cenário chegou-se a conclusão que entre quarto e cinco nós simulando algo em torno de 280/350 *threads* (requisições) é o quanto a arquitetura *ispTV* comporta a um nível interessante de tempo de resposta e de integridade das requisições.

O segundo cenário segue o mesmo processo para realização do teste. O que difere do primeiro é a capacidade de processamento de mais requisições e o suporte em alto nível a mais nós. Neste cenário, o servidor em que a arquitetura está presente começou a apresentar um *delay* mais significativo quando chegou perto do sétimo nó, o que demonstra que foi capaz de atender algo em torno de 490 requisições com um tempo de resposta entre 2300/2500 ms.

Semelhante ao segundo cenário, justificando pela semelhança de configuração dos servidores, no terceiro cenário foi possível identificar alterações na qualidade das respostas ao iniciar o oitavo e último nó, representando o suporte à execução em alta qualidade de aproximadamente 560 requisições por oito nós simultaneamente com tempo de resposta entre 2250/2350.

## 5.4 Considerações Finais

A conclusão que pode ser obtida após a análise dos três cenários é que a arquitetura diante os três servidores utilizados, em média, suporta em torno de 455 requisições simultaneamente atendendo as requisições de forma que a execução da

aplicação não seja prejudicada. Todas as requisições atendidas foram processadas de forma correta e retornaram o resultado da forma esperada.

O número de 455 requisições satisfaz o objetivo deste trabalho em atender a demanda de utilização desta arquitetura em emissoras de pequeno porte e em auxiliar desenvolvedores e pesquisadores que necessitam de utilizar uma arquitetura de PSI utilizando-o no canal de interatividade para a execução das aplicações interativas. Para uma transmissão em grande escala dos programas televisivos e a possibilidade de ter milhares de requisições por segundo a um serviço, deve-se pensar em uma organização de distribuição de recursos para que a arquitetura *ispTV* garanta a capacidade de oferecimento e consistência das informações no cenário do SBTVD. Com essa distribuição a capacidade da arquitetura deve aumentar gradativamente diante as técnicas utilizadas para a distribuição dos recursos, como Clustering (CARDELLINI, V. *et al*, 2001). Vale lembrar que o objetivo é atender a grande demanda de requisições que pode ocorrer em um país de grande dimensão, como o Brasil, para que não seja afetada a qualidade de resposta de cada serviço solicitado.

A análise que pode ser feita é sobre a distribuição da arquitetura em *clusters* ou grades para diminuir o tempo de resposta, distribuir melhor os recursos que conseqüentemente distribuirão as requisições e aumentará a capacidade de atendimento de requisições.

Como trabalho futuro para testes de desempenho e desenvolvimento, a distribuição da arquitetura pode ser feita de forma simples e independente. Cada nó presente no *cluster*, ou em uma arquitetura de grades, possuiria uma arquitetura *ispTV* que seria gerenciada por um nó principal de escalonamento de requisição. Estes são alguns dos passos para trabalhos futuros relacionados à arquitetura.

# Capítulo 6

## CONCLUSÃO

---

*Este capítulo apresenta as conclusões baseadas no desenvolvimento da arquitetura ispTV, aplicações de estudo de caso, nos testes realizados para análise da capacidade da arquitetura e em trabalhos futuros.*

Para prover os sistemas de TVDI da capacidade de transmissão de programas interativos que possuem a necessidade de armazenamento das informações resultantes da interação, o canal de interatividade deve ser utilizado junto a um PSI capaz de concentrar as informações necessárias. Junto a estas características dos sistemas de TVDI, foi apresentado que o SBTVD garante que os programas interativos produzidos tenham o caráter de inclusão social, cultural e tecnológico. Diante o SBTVD, este trabalho apresentou uma arquitetura interativa genérica de um PSI baseado em uma arquitetura orientada a serviços para oferecer a possibilidade de que os programas interativos desenvolvidos no SBTVD se comuniquem com a Emissora através do canal de interatividade pelo PSI (*ispTV*).

A arquitetura apresentada oferece aos sistemas de TVDI funcionalidades que um PSI não genérico e que não utiliza como padrão diversas tecnologias de serviços para publicação não podem oferecer. Diante das características apresentadas, pode-se destacar a API cliente oferecida como forma de abstrair para o desenvolvedor toda a parte de comunicação com a arquitetura e se preocupar apenas com os dados a serem enviados. Do lado da Emissora, a arquitetura oferece a facilidade de criação de serviços, os *ispTV services*, baseados em apenas classes *Java*, o que permite que os desenvolvedores de aplicações interativas não tenham conhecimento das tecnologias necessárias para desenvolver serviços web. E para finalizar, do lado do canal de interatividade, mais especificamente do PSI, uma aplicação Web de gerenciamento do serviços presentes no PSI e diferentes formas de publicação destes serviços.

Analisando a arquitetura de modo geral inserida no contexto do SBTVD pode-se chegar a conclusão que a facilidade de gerenciamento, manutenção e acesso às funcionalidades são as maiores vantagens da *ispTV*. Nos sistemas de TVDI é uma grande vantagem a utilização da arquitetura pela modificação constante dos conteúdos dos programas televisivos, em que as aplicações envolvidas nos programas estão se modificando para atender o conteúdo a ser transmitido. Utilizando a arquitetura *ispTV*, as emissoras ganham tempo desenvolvendo os serviços web específicos para suas aplicações interativas pelas formas de publicação oferecidas pela arquitetura e a possibilidade de utilização de serviços remotos.

Pode-se concluir também que devido à padronização de mensagem utilizada na arquitetura, é possível torna-la independente das tecnologias utilizadas no *middleware* brasileiro. Isso pelo fato de que a arquitetura é baseada na estrutura da mensagem enviada para ela e não na tecnologia utilizada para enviar a mensagem. Isto possibilita que diversas aplicações interativas, não só as interativas, acessem a arquitetura para obter informações dos serviços específicos publicados, possibilitando uma maior abrangência na sua utilização.

Nos testes apresentados demonstrou-se uma análise da arquitetura envolvendo testes de usabilidade e testes de desempenho, envolvendo teste de requisições e escalabilidade. Analisando o teste de usabilidade, chegou-se a conclusão que as APIs de comunicação facilitam a forma de comunicação com a arquitetura *ispTV*. Ao analisar os testes, pode-se obter um número de requisições que a arquitetura suporta para certos cenários e de que atende as motivações e objetivos destacados neste trabalho. As motivações são atendidas por possibilitar que emissoras de pequeno porte e desenvolvedores possam desenvolver aplicações interativas de inclusão social e disseminação da informação utilizando a comunicação através do canal de interatividade. Já os objetivos são alcançados por conseguirem oferecer com qualidade de comunicação, a publicação de serviços interativos e oferecer a comunicação de forma genérica das aplicações interativas com os serviços interativos desenvolvidos.

Diante o cenário global do SBTVD seria de extrema importância a utilização de clusterização para compartilhamento dos recursos necessários para o atendimento de milhares de requisições.

Com isso, pode-se chegar à conclusão que a arquitetura proposta prega conceitos de genericidade e de simples usabilidade visando facilitar o processo de criação dos programas interativos. Isto é de muita importância para o SBTVD para que a evolução do *middleware* Ginga seja crescente e para que as aplicações evoluam cada vez mais para proporcionar melhores experiências aos usuários. Do ponto de vista de desempenho, a arquitetura sendo gerenciada por apenas um servidor não possui capacidade de abrangência de todo o território brasileiro. Isto não significa que seu funcionamento está abaixo do esperado, mas sim que a demanda para aplicações interativas será além de sua capacidade de resposta.

A necessidade de se obter maior desempenho e escalabilidade para oferecer mais qualidade para as requisições das aplicações interativas deixa possibilidades de trabalhos futuros para esta arquitetura. Este trabalho se aplicaria na organização de N arquiteturas em um cluster ou grid computacional, ou mesmo, melhorando a organização interna de seus serviços para que eles possam ser replicados em destinos diferentes e o *Broker* seja o responsável não só apenas pelo redirecionamento, mas também, pelo reescalonamento das requisições. São diversas abordagens que podem ser utilizadas para aumentar-se a capacidade de desempenho da arquitetura e, neste contexto, é sugerido um trabalho futuro para a arquitetura *ispTV*.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

---

ABNT NBR 15607-1: Norma para Televisão Digital Terrestre - Canal de Interatividade - Parte 1: Protocolos, Interfaces Físicas e Interfaces de Software. Disponível em: <http://www.forumsbtvd.org.br/materias.asp?id=112>. Acesso em: 23 de Março de 2011.

ATSC - Padrão Americano de Televisão Digital. Disponível em: <http://www.atsc.org/cms>. Acesso em: 24 de Novembro de 2010.

ÁVILA, P. M. D.; ZORZO, S. D. A personalized TV Guide System. *SMC*, p. 2122 - 2127, 2009.

Axis2 Web Services/SOAP/WSDL engine. Disponível em: <http://axis.apache.org/axis2/java/core/>. Acesso em: 05 de Janeiro de 2011.

BAKMAZ, B.; BOJKOVIC, Z. Trends in Interactive Services for Digital Television Infrastructure. *WSEAS*, v. 7, n. 12, 2008.

CAMPISTA, M. E. M. et al. The Ad Hoc Return Channel: A Low-Cost Solution for Brazilian Interactive Digital TV. *IEEE Communications Magazine*, p. 136-143, 2007.

CARDELLINI, V.; CASALICCHIO, E.; COLAJANNI, M. A performance study of distributed architectures for the quality of Web services, *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, vol., no., pp. 10 pp., 3-6 Jan. 2001

CESAR, P.; BULTERMAN, D. C. A.; SOARES, L. F. G. Introduction to Special Issue: Human-Centered Television - Directions in Interactive Digital Television Research. *ACM Trans. Multimed. Comput. Comm. Appl*, v. 4, 2008.

*Decreto Brasileiro n° 4.901, 2003.* Disponível em: [http://sbtvd.cpqd.com.br/downloads/decreto\\_4901\\_2003.pdf](http://sbtvd.cpqd.com.br/downloads/decreto_4901_2003.pdf). Acesso em: 14 de Fevereiro de 2011.

*DVB - Padrão Europeu de Televisão Digital.* Disponível em: <http://www.dvb.org>. Acesso em: 23 de Novembro de 2010.

ERL, T. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design.* Prentice Hall, 2005.

FERNANDES, J.; LEMOS, G.; SILVEIRA, G. *Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas.* JAI-SBC, 2004.

GEERTS, D.; CESAR, P.; BULTERMAN, D. *The Implications of Program Genres for the Design of Social Television Systems.* UXTV, 2008.

IBGE; PNAD. *Pesquisa Nacional de Amostra de Domicílios (PNAD) - TIC (Telefones Fixos e Celulares, Microcomputadores, Internet, Rádio e Televisão).* 2009. Disponível em: <http://www.teleco.com.br/pnad.asp>. Acesso em: 08 de Dezembro de 2010.

*ISDB-T - Padrão Japonês de Televisão Digital.* Disponível em: <http://www.dibeg.org>. Acesso em: 15 de Fevereiro de 2011.

*Java.* Disponível em: <http://www.oracle.com/technetwork/java/index-jsp-142903.html#documentation>. Acesso em: 05 Janeiro de 2011

*LUA.* Disponível em: <http://www.lua.org>. Acesso em: 05 de Janeiro de 2011.

MANHÃES, M. A. R.; P. J. Shieh.; et al. *Canal de Interatividade em TV Digital.* Caderno CPqD Tecnologia, v. 1, 2005.

MENG, J; MEI, S; YAN, Z. RESTful Web Services: A Solution for Distributed Data Integration. *Computational Intelligence and Software Engineering, CiSE. International Conference on*, pp.1-4, 11-13, Dec. 2009.

MONTEZ, C.; BECKER, V. TV Digital Interativa: Conceitos, Desafios e Perspectivas para o Brasil. 2 ed. Editora da UFSC, 2005.

NCL - *Nested Context Language*. Disponível em: <http://www.ncl.org.br>. Acesso em: 05 de Janeiro de 2011.

*Pedra do Ingá*. Disponível em: [http://www.viafanzine.jor.br/fonseca\\_inga.htm](http://www.viafanzine.jor.br/fonseca_inga.htm). Acessado em: 05 Janeiro de 2011

PICCOLO, L. S. G.; BARANAUSKAS, M. C. C. Understanding iDTV in a Developing Country and Designing a T-gov Application Prototype. *DIS*, 2008.

SANTOS JUNIOR, J. B. A., I. C.; LIMA, A, N. B.; LOIOLA, M. A.; ÁVILA, P. M.; MARQUES, H. N.; KUEHNE, B. T. Prototyping Interactive Digital Television Programs Using Java and XML: A Platform for Developers and Viewers. *EuroITV*, 2007.

SANTOS JUNIOR, J. B. A., I.C. ; MORSELLI JUNIOR, J. C. M. ; TEIXEIRA, F. C.; PRADO, G. M. ; ÁVILA, P. M. Back Channel in Interactive Digital Television Systems: Strategies for Prototyping Applications Using an Interactive Service Provider. *ICEIS*, v. 1, 2009.

SEOK, J. M.; CHOI, J. H.; CHOI, J. S. The Design of Web-based Return Channel System in the Data Broadcast Services . *ICACT*, v. 2, 2005.

SOARES, L. F. G.: TV Interativa se faz com Ginga. Disponível em: [http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009\\_06\\_soares.pdf](http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/2009_06_soares.pdf). Acesso em: 06 de Janeiro de 2011.

SOARES, L. F. G.; RODRIGUES, R. F.; MORENO, M. F. Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. *JBCS*, 2007.

SUZUMURA, T.; TRENT, S.; TATSUBORI, M.; TOZAWA, A.; ONODERA, T.; , Performance Comparison of Web Service Engines in PHP, Java and C, *Web Services, 2008. ICWS '08. IEEE International Conference on* , vol., no., pp.385-392, 23-26 Sept. 2008

*XPTA: XPTA.LAB*. Disponível em: <http://www.culturadigital.br/xpta/>. Acessado em: 27 de Setembro de 2011.

*JAX-WS Reference Implementation Project*. Disponível em: <http://jax-ws.java.net/>. Acesso em: 05 de Janeiro de 2011.