



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**XTATION: UM AMBIENTE PARA EXECUÇÃO E
TESTE DE APLICAÇÕES INTERATIVAS PARA O
MIDDLEWARE GINGA**

FERNANDA PAULINELLI RODRIGUES SILVA

JOÃO PESSOA/PB

JULHO DE 2010



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**XTATION: UM AMBIENTE PARA EXECUÇÃO E
TESTE DE APLICAÇÕES INTERATIVAS PARA O
MIDDLEWARE GINGA**

FERNANDA PAULINELLI RODRIGUES SILVA

JOÃO PESSOA/PB

JULHO DE 2010

FERNANDA PAULINELLI RODRIGUES SILVA

**XTATION: UM AMBIENTE PARA EXECUÇÃO E
TESTE DE APLICAÇÕES INTERATIVAS PARA O
MIDDLEWARE GINGA**

Dissertação apresentada ao Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, como requisito parcial para obtenção do título de Mestre em Informática (Sistemas de Computação).

Orientação: Prof. Dra. Tatiana Aires Tavares

JOÃO PESSOA/PB

JULHO DE 2010

S586x Silva, Fernanda Paulinelli Rodrigues.

Xtation: um ambiente para execução e teste de aplicações interativas para o
Middleware Ginga / Fernanda Paulinelli Rodrigues Silva. - - João Pessoa: [s.n.], 2010.
93 f. : il.

Orientador: Tatiana Aires Tavares.
Dissertação (Mestrado) – UFPB/CCEN.

1.Sistemas de Computação. 2.TV Digital Interativa. 3.Aplicações interativas.
4.Middleware Ginga -Testes.

UFPB/BC

CDU:004 (043)

1

Ata da Sessão Pública de Defesa de Dissertação de
Mestrado de FERNANDA PAULINELLI
RODRIGUES SILVA, candidata ao Título de
Mestre em Informática na Área de Sistemas de
Computação, realizada em 05 de julho de 2010.

2

3

4

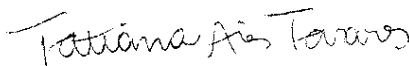
5 Aos cinco dias do mês de julho do ano dois mil e dez, às quatorze horas, na Sala de
6 Reunião do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba,
7 reuniram-se os membros da Banca Examinadora constituída para examinar a candidata ao
8 grau de Mestre em Informática, na área de “*Sistemas de Computação*”, na linha de pesquisa
9 “*Computação Distribuída*”, a Sra. Fernanda Paulinelli Rodrigues Silva. A comissão
10 examinadora composta pelos professores doutores: Tatiana Aires Tavares (DI - UFPB),
11 Orientador e Presidente da Banca Examinadora, Dênio Mariz Timóteo de Sousa (DI-
12 UFPB), como examinador interno e Aquiles Burlamarqui (UFRN) como examinador
13 externo. Dando início aos trabalhos, a Prof^a. Tatiana Aires Tavares, cumprimentou os
14 presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra a candidata
15 para que a mesma fizesse, oralmente, a exposição do trabalho de dissertação intitulado
16 “*XTATION: UM AMBIENTE PARA EXECUÇÃO E TESTE DE APLICAÇÕES*
17 *INTERATIVAS PARA O MIDDLEWARE OPENINGA*”. Concluída a exposição, a
18 candidata foi argüida pela Banca Examinadora que emitiu o seguinte parecer: “*aprovado*”.
19 Assim sendo, deve a Universidade Federal da Paraíba expedir o respectivo diploma de
20 Mestre em Informática na forma da lei e, para constar, o professor Lucídio dos Anjos
21 Formiga Cabral, Sr. Vice-Coordenador do PPGI, lavrou a presente ata, que vai assinada
22 por ela, e pelos membros da Banca Examinadora. João Pessoa, 05 de julho de 2010.

23

24

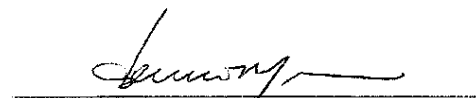

Lucídio dos Anjos Formiga Cabral

Prof^a. Dr^a. Tatiana Aires Tavares
Orientadora (DI-UFPB)



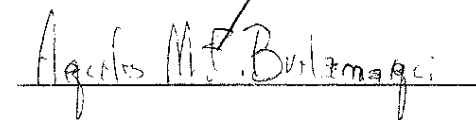
Tatiana Aires Tavares

Prof. Dr. Dênio Mariz Timóteo de Sousa
Examinador Interno (DI-UFPB)



Dênio Mariz Timóteo de Sousa

Prof. Dr. Aquiles Burlamarqui
Examinador Externo (UFRN)



Aquiles M.F. Burlamarqui

25

Dedico este trabalho ao meu esposo Vinicius, verdadeiro companheiro, por tamanha dedicação e amor;

Aos meus pais, Fernando e Maria Luiza, por não medirem esforços para me oferecer uma boa educação e por todo amor, atenção e ensinamentos que sempre me dedicaram.

AGRADECIMENTOS

A Deus, por me permitir alcançar tantas conquistas;

A minha orientadora, Tatiana, pelo apoio;

Aos meus mestres, por todas as oportunidades e pelo incentivo durante os anos de aprendizado no Departamento de Informática;

Aos amigos do LAViD dos velhos tempos de formação do laboratório, pelos anos de convivência e “demonstrações”;

Ao Marcovas e ao Alan, pela disponibilidade e ajuda;

A todos que, direta ou indiretamente, contribuíram com o sucesso deste trabalho.

RESUMO

Montar uma infraestrutura para a realização de testes reais em aplicações para TV Digital Interativa (TVDI) envolve o uso de tecnologias de ponta e, conseqüentemente, depende de um alto investimento em uma infraestrutura que simule o funcionamento de uma Rede de Televisão. Este investimento se coloca como uma barreira tecnológica que impede a disseminação do conhecimento no desenvolvimento e teste de programas para TVDI. De forma a contornar este obstáculo, é apresentado neste trabalho o desenvolvimento do Xtation, um ambiente para execução e testes de aplicações de TVDI para o Ginga, o middleware de referência do Sistema Brasileiro de Televisão Digital (SBTVD). Através de uma arquitetura distribuída composta por um conjunto de ferramentas controladas remotamente por uma aplicação de interface simples e intuitiva, o Xtation tem como principal atrativo seu baixo custo ao permitir a realização de testes confiáveis de aplicações interativas, mediante a simulação de um ambiente de TVDI real.

Palavras-chave: TV Digital Interativa; Aplicações Interativas; Testes; Ginga.

ABSTRACT

Providing an infrastructure for the accomplishment of real tests in Applications for Interactive Digital TV (IDTV) involves the use of avant-garde technologies and, consequently, it depends on a high investment in an infrastructure that simulates the operation of a Television Network. This investment is a technological barrier that restrains the knowledge dissemination in the development and test of IDTV programs. In order to overcome this problem, this work proposes the development of Xtation, an environment for execution and test of IDTV applications based on Ginga, the reference middleware of the Brazilian Digital Television System (SBTVD). Through a distributed architecture composed of a set of tools remotely controlled by a simple and intuitive application, the Xtation's main attraction is its low cost to allow for reliable testing of interactive applications by simulating a real IDTV environment.

Keywords: Digital and Interactive TV, Interactive Applications; Tests; Ginga

LISTA DE ILUSTRAÇÕES

FIGURA 1: COMPARATIVO ENTRE TV TRADICIONAL E TV INTERATIVA	26
FIGURA 2: MODELO DE UM SISTEMA DE TVDI	28
FIGURA 3: ETAPAS DA DIFUSÃO DO CONTEÚDO TELEVISIVO	29
FIGURA 4: ARQUITETURA EM CAMADAS DE UM RECEPTOR DE TVDI.....	30
FIGURA 5: ARQUITETURA EM ALTO NÍVEL DO MIDDLEWARE GINGA.....	33
FIGURA 6: CONJUNTO DE APIS QUE COMPÕE A ESPECIFICAÇÃO DO GINGA-J.....	34
FIGURA 7: CONJUNTO DE APIS JAVA DTV.	35
FIGURA 8: CICLO DE VIDA DAS APLICAÇÕES DE TVDI.....	37
FIGURA 9: EXEMPLOS DE APLICATIVOS SEM RELAÇÃO COM CONTEÚDO AUDIOVISUAL DO PROGRAMA	37
FIGURA 10: PROTÓTIPO DE PROGRAMA DE TVDI COM APLICAÇÃO FORTEMENTE ACOPLADA.....	38
FIGURA 11: EXEMPLO APLICAÇÃO PARA A CATEGORIA DE NOTÍCIAS	39
FIGURA 12: EXEMPLO DE APLICAÇÃO NA CATEGORIA ESPORTIVA.....	39
FIGURA 13: EXEMPLO DE APLICAÇÃO PARA AS ELEIÇÕES.....	39
FIGURA 14: EXEMPLO DE APLICATIVO SEM MOMENTO DEFINIDO PARA EXECUÇÃO	40
FIGURA 15: EXEMPLO DE APLICAÇÃO DESENVOLVIDA EM NCL.....	41
FIGURA 16: PROTÓTIPO DE PROGRAMA DE TVDI PARA UM JORNAL INTERATIVO	42
FIGURA 17: EXEMPLO DE APLICAÇÃO DE INTERATIVIDADE LOCAL, INTERMITENTE E PERMANENTE	43
FIGURA 18: RESUMO DAS PRINCIPAIS CARACTERÍSTICAS DAS APLICAÇÕES E PROGRAMAS DE TVDI.....	44
FIGURA 19: PROCESSO DE DESENVOLVIMENTO DE PROGRAMAS PARA TVDI.....	45
FIGURA 20: FASES DO MODELO DE PROCESSO PARA DESENVOLVIMENTO DE PROGRAMAS DE TVDI	47
FIGURA 21: TELA PRINCIPAL DO XLETVIEW.....	53
FIGURA 22: GERENCIADOR DE APLICAÇÕES DO XLETVIEW.....	54
FIGURA 23: GERENCIADOR DE APLICAÇÕES DO OPENMHP.....	55
FIGURA 24: TELA DE PROPRIEDADES DA APLICAÇÃO NO OPENMHP.....	55
FIGURA 25: JANELAS DE SAÍDA GRÁFICA E TEXTUAL DO OPENMHP.....	55
FIGURA 26: TESTE DE APLICAÇÃO NCL NO GINGA-NCL EMULADOR	56
FIGURA 27: INTERFACE GRÁFICA DO STB VIRTUAL GINGA-NCL.....	57
FIGURA 28: TESTE DE APLICAÇÃO NO STB VIRTUAL GINGA-NCL	57
FIGURA 29: INTERFACE DO IRT MIDDLEWARE	58
FIGURA 30: INTERFACE DO ITR TEST CENTRE	59
FIGURA 31: PARTE DA INFRAESTRUTURA UTILIZADA PELO IRT MHP TEST CENTRE	59
FIGURA 32: ARQUITETURA DO AMBIENTE XTATION.....	63
FIGURA 33: ATIVIDADES DO PROCESSO DE PRODUÇÃO DE PROGRAMAS DE TVDI ONDE O XTATION ESTÁ INSERIDO.	66
FIGURA 34: APLICABILIDADE DO XTATION NO PROCESSO DE PRODUÇÃO DE PROGRAMAS DE TVDI	66
FIGURA 35: DIAGRAMA DE CASOS DE USO UML DO XTATION	68
FIGURA 36: DIAGRAMA DE PACOTES DO XTATION	72
FIGURA 37: DIAGRAMA DE CLASSES DO XTATION	73
FIGURA 38: DIAGRAMA DE IMPLANTAÇÃO DO XTATION	74
FIGURA 39: TELA DO INICIAL DO XTATIONWIZARD COM AS OPÇÕES "NOVO PROJETO" E "CONFIGURAR"	75
FIGURA 40: TELA DE CONFIGURAÇÃO DO AMBIENTE XTATION	75
FIGURA 41: TELA DO XTATIONWIZARD QUE COLETA AS INFORMAÇÕES SOBRE A APLICAÇÃO E O VÍDEO.....	75
FIGURA 42: TELA DO XTATIONWIZARD PARA ACOMPANHAMENTO DO PROCESSO DE TESTE	76

FIGURA 43: EXEMPLO DE TELA DE LOG DE EXECUÇÃO DOS PROGRAMAS INTERATIVOS.....	76
FIGURA 44: INFRAESTRUTURA UTILIZADA NA VALIDAÇÃO DO XTATION.....	78
FIGURA 45: CASOS DE USO UML PARA A TORCIDA VIRTUAL	80
FIGURA 46: ESTUDO DE CASO 1: UTILIZAÇÃO DO XTATION NO AUXÍLIO À ATIVIDADE DE PROGRAMAÇÃO VISUAL	81
FIGURA 47: ESTUDO DE CASO 2: UTILIZAÇÃO DO XTATION NO AUXÍLIO À ATIVIDADE DE CODIFICAÇÃO DO SOFTWARE	81
FIGURA 48: ESTUDO DE CASO 3: UTILIZAÇÃO DO XTATION NO AUXÍLIO À ATIVIDADE DE INTEGRAÇÃO SOFTWARE/PROGRAMA TV	82
FIGURA 49: ESTUDO DE CASO 4: UTILIZAÇÃO DO XTATION NO AUXÍLIO À ATIVIDADE DE TESTES	83
FIGURA 50: LOG DE EXECUÇÃO DA TORCIDA VIRTUAL	84

LISTA DE TABELAS

TABELA 1: DEFINIÇÕES DE TV INTERATIVA ENCONTRADAS NA LITERATURA	24
TABELA 2: ESTÁGIOS DA PRODUÇÃO TELEVISIVA (GAWLINSKI, 2003).....	46
TABELA 3: DIVERSOS TESTES UTILIZADOS NA PRODUÇÃO DE PROGRAMAS DE TVDI	48
TABELA 4: EQUIPES ENVOLVIDAS NO PROCESSO DE PRODUÇÃO DE PROGRAMAS DE TVDI.	50
TABELA 5: REQUISITOS FUNCIONAIS DO XTATION.....	69
TABELA 6: REQUISITOS NÃO-FUNCIONAIS DO XTATION.....	69
TABELA 7: CASO DE USO EXPANDIDO: CONFIGURAR O AMBIENTE	70
TABELA 8: CASO DE USO EXPANDIDO: INSERIR INFORMAÇÕES SOBRE O VÍDEO	71
TABELA 9: CASO DE USO EXPANDIDO: INSERIR INFORMAÇÕES SOBRE A APLICAÇÃO	71
TABELA 10: CASO DE USO EXPANDIDO: TESTAR A APLICAÇÃO.....	71
TABELA 11: ESPECIFICAÇÕES DOS EQUIPAMENTOS UTILIZADOS NA INFRAESTRUTURA DO XTATION	78

LISTA DE ABREVIATURAS E SIGLAS

ACAP	<i>Advanced Common Application Platform</i>
AIT	<i>Application Information Table</i>
API	<i>Application Programming Interfaces</i>
ARIB	<i>Association of Radio Industries and Businesses</i>
ATSC	<i>Advanced Television Systems Committee</i>
CTIC	Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação
DASE	<i>Digital TV Applications Software Environment</i>
DAVIC	<i>Digital Audio Video Council</i>
DSM-CC	<i>Digital Storage Media Command and Control Protocol</i>
DVB	<i>Digital Video Broadcast</i>
EPG	<i>Electronic Program Guide</i>
GEM	<i>Globally Executable MHP</i>
GINGA-CC	<i>Ginga Common Core</i>
GINGA-CDN	<i>Ginga Code Development Network</i>
GPL	<i>General Public License</i>
HAVi	<i>Home Audio Video Interoperability</i>
HDTV	<i>High Definition Television</i>
IRT	<i>Institut für Rundfunktechnik GmbH</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
ITU	<i>International Telecommunication Union</i>
JMF	<i>Java Media Framework</i>
JVM	<i>Java Virtual Machine</i>
LAViD	Laboratório de Aplicações de Vídeo Digital
LGPL	<i>Lesser General Public Licence</i>
LWUIT	<i>LightWeight User Interface Toolkit</i>
MHP	<i>Multimedia Home Platform</i>

MHP-KDB	<i>MHP Knowledge Database</i>
NCL	<i>Nested Context Language</i>
OC	<i>Object Carousel</i>
OCAP	<i>OpenCable Application Platform</i>
RMI	<i>Remote Method Invocation</i>
RNP	Rede Nacional de Ensino e Pesquisa
RPC	<i>Remote Procedure Call</i>
SBTVD	Sistema Brasileiro de Televisão Digital
SBTVD-T	Sistema Brasileiro de Televisão Digital Terrestre
SI	<i>Service Information</i>
SSH	<i>Secure Shell</i>
STB	<i>Set-top Box</i>
TS	<i>Transport Stream</i>
TV	Televisão
TVD	TV Digital
TVDI	TV Digital Interativa
TVe	TV Expandida
TVi	TV Interativa
UML	<i>Unified Modeling Language</i>
XP	<i>eXtreme Programming</i>

SUMÁRIO

INTRODUÇÃO	16
1.1 MOTIVAÇÃO	18
1.2 TRABALHOS CORRELATOS NA INSTITUIÇÃO	19
1.3 OBJETIVOS	21
1.3.1 Objetivo Geral	21
1.3.2 Objetivos Específicos.....	21
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO	21
FUNDAMENTAÇÃO TEÓRICA.....	23
2.1. A TV DIGITAL INTERATIVA	23
2.1.1 COMPONENTES DA TV DIGITAL INTERATIVA	27
2.2 O MIDDLEWARE GINGA E A IMPLEMENTAÇÃO OPENGINGA	32
2.3 APLICAÇÕES E PROGRAMAS INTERATIVOS PARA TVDI.....	36
2.4 PROCESSO DE PRODUÇÃO DE PROGRAMAS INTERATIVOS PARA TVDI.....	44
2.5 DISCUSSÃO	51
TRABALHOS RELACIONADOS.....	52
3.1 XLETVIEW	52
3.2 OPENMHP.....	54
3.3 GINGA-NCL EMULADOR E SET-TOP BOX VIRTUAL GINGA-NCL	56
3.4 IRT MIDDLEWARE E IRT MHP TEST CENTRE	57
3.5 ANÁLISE COMPARATIVA DAS FERRAMENTAS.....	59
O AMBIENTE XTATION	62
4.1 VISÃO GERAL	62
4.2 COMPONENTES	63
4.2.1 Streamer.....	63
4.2.2 Gerador de Carrossel.....	64
4.2.3 Remux - Multiplexador.....	64
4.2.4 Servidor de Middleware.....	64
4.2.5 XtationWizard.....	65
4.3 APLICABILIDADE	65
4.4 ANÁLISE E PROJETO DO XTATION	67

4.4.1 Especificação de Requisitos e Visão de Casos de Uso.....	68
4.4.2 Expansão dos Casos de Uso	70
4.4.3 Projeto da Arquitetura – Visão Lógica.....	72
4.4.4 Visão de Implementação.....	72
4.4.5 Visão de Implantação	73
4.4.6 Projeto da Interface com o Usuário	74
ESTUDO DE CASO	77
5.1 A INFRAESTRUTURA	77
5.2 O PROGRAMA DE TVDI.....	79
5.3 UTILIZAÇÃO DO XTATION NO PROCESSO DE PRODUÇÃO DA TORCIDA VIRTUAL.....	80
CONSIDERAÇÕES FINAIS.....	85
6.1 HISTÓRICO E DIFICULDADES ENCONTRADAS.....	86
6.2 OPORTUNIDADES	87
REFERÊNCIAS BIBLIOGRÁFICAS	89

1

INTRODUÇÃO

A chegada da tão esperada interatividade na Televisão Digital traz consigo uma mudança significativa na forma de produzir e realizar programas televisivos. A grande novidade está na possibilidade de veiculação de softwares interativos, também chamados de aplicações, juntamente com os fluxos de áudio e vídeo transmitidos pelas emissoras, fluxos estes que na TV Digital vêm com uma qualidade bastante superior em relação às transmissões analógicas. Diante desta inovação, surge uma grande demanda para produção e desenvolvimento de serviços e aplicações dos mais diversos tipos, voltados para esta nova mídia interativa. Durante este processo de produção, uma das fases que merece grande atenção é a de testes.

O processo de teste de programas para televisão tradicional se resume basicamente à verificação da existência de problemas causados por *videotapes* danificados, níveis de luz inadequados ou erros introduzidos na edição. Não há necessidade de muitos testes, pois é muito pouco provável que o programa em si cause quaisquer problemas com os equipamentos de TV do telespectador ou com o canal de televisão, já que as falhas tendem a acontecer dentro dos próprios programas, durante sua gravação, e não no momento da entrega da tecnologia (GAWLINSKI, 2003).

Em contraste, a maioria dos softwares para computador consiste em um conjunto de componentes complexos que precisam trabalhar sob uma série de circunstâncias específicas, de modo que estes softwares precisam ser submetidos a processos rigorosos de testes. Ainda sim, a despeito da realização de diferentes tipos de verificações, os computadores dão problemas, os softwares por vezes não funcionam como esperado e, mesmo quando

funcionam, muitos são de difícil utilização. Um fato interessante é que os usuários de computador já estão, de certa forma, acostumados com estas dificuldades.

Teste completo não é possível, mas mesmo assim esta fase é considerada um elemento crítico da garantia de qualidade de software e de conformidade com os requisitos especificados (PRESSMAN, 2002). No processo de desenvolvimento de aplicações para TV Digital Interativa não seria diferente. Como estas aplicações nada mais são do que softwares é fundamental que elas sejam bem testadas, visto que muitas vezes o alcance que as mesmas terão ao serem disponibilizadas será da ordem de milhões de televisores com os mais diversos tipos de usuários ao redor do mundo. Diga-se de passagem, estes usuários não iriam gostar nem um pouco da idéia de reiniciar constantemente seus aparelhos de TVDI devido a problemas técnicos dos serviços interativos. Além do mais, o atendimento a estes problemas podem trazer altos custos aos provedores destes serviços.

O problema com as aplicações de televisão interativa é que elas têm toda a complexidade das aplicações de computadores, mas estão rodando em uma plataforma que normalmente não tem problemas técnicos ou de usabilidade. Os aparelhos de televisão dos telespectadores não falham, o controle remoto sempre funciona (a menos que a bateria acabe) e, ao contrário da Internet, a imagem não carrega mais lentamente nos horários de pico de visualização. Qualquer serviço de televisão interativa que cause problemas com o aparelho de televisão ou faça coisas inesperadas provavelmente não será tolerado pelos telespectadores - especialmente se eles estão experimentando a televisão interativa pela primeira vez e ainda estão aprendendo sobre aquilo que ela tem para oferecer. Eles não vão querer voltar para mais castigo. (GAWLINSKI, 2003 p. 173), tradução nossa.

Em uma fase preliminar de testes podem ser utilizados emuladores, que nada mais são que programas que simulam a execução da aplicação de TV no computador. Porém, por não garantir o correto funcionamento do programa interativo em um ambiente real, a utilização de emuladores é insuficiente, devido, entre outras limitações, às diferenças significativas no desempenho e na interface com o usuário. Por outro lado, simular o funcionamento de uma rede de televisão para a realização de testes reais em aplicações para TV Digital Interativa envolve o uso de equipamentos que exigem um alto investimento na montagem de uma infraestrutura adequada.

Portanto, a fim de garantir aplicações interativas adequadas ao formato da TVDI, torna-se imprescindível adicionar ao processo de desenvolvimento uma ferramenta que reproduza as características relevantes de um ambiente de TV e possibilite a realização de testes confiáveis. É neste contexto que se encaixa o foco deste trabalho.

O Xtation tem como principal objetivo permitir a execução e testes de programas para TVDI em um ambiente simples e de baixo custo, mas que permita a simulação de um ambiente de TVDI real e auxilie o processo de produção de aplicações e programas interativos. Ao testar aplicações interativas estamos, por consequência, validando a implementação do *middleware* utilizado, de forma que o Xtation também é de grande valia para o teste de implementações do *middleware* Ginga.

1.1 MOTIVAÇÃO

A motivação para a realização deste trabalho surgiu no LAViD¹ (Laboratório de Aplicações de Vídeo Digital da Universidade Federal da Paraíba) através de alguns anos de participação em projetos na área de TV Digital, mais especificamente no desenvolvimento e teste de aplicações interativas.

Surgiu então a necessidade de desenvolvimento de uma ferramenta que permitisse a execução e realização de testes das aplicações desenvolvidas no laboratório. Os emuladores existentes no mercado já não satisfaziam nossas necessidades. Primeiro por possuírem implementação apenas parcial das bibliotecas utilizadas para a construção das aplicações; segundo porque existem diferenças consideráveis entre testar aplicações no computador e na televisão, principalmente pelo ponto de vista da apresentação da interface. Por sua vez, as ferramentas comerciais, apesar de serem mais completas e confiáveis, possuem alto custo na aquisição de suas licenças.

Portar uma aplicação testada apenas no computador, depois de pronta, para um ambiente de televisão resulta em muitas horas de trabalho devido às características: (a) dos receptores, como a capacidade de armazenamento e velocidade; (b) das televisões, como resolução, diferenças no tamanho dos pixels (quadrados no computador e levemente

¹ <http://www.lavid.ufpb.br>

retangulares na televisão), tamanho das fontes e diferenças na saturação, brilho e contraste das cores; e (c) próprias da utilização das aplicações pelos usuários, como maior distanciamento da tela e utilização do controle remoto como dispositivo de interação.

Estas dificuldades se colocam como uma barreira tecnológica que impede a disseminação do conhecimento no desenvolvimento e teste de programas para TVDI. A motivação para a realização deste trabalho é justamente a proposta de desenvolvimento e divulgação de uma alternativa de baixo custo e confiável que, ao ser incorporada no processo de produção de conteúdo para TVDI, garanta um produto de qualidade na televisão dos telespectadores.

Corroborando com a realização deste trabalho, o LAViD tem participação ativa no processo de especificação do *middleware* de referência do Sistema Brasileiro de TV Digital, o Ginga. Além disto, vem sendo desenvolvida também neste laboratório uma implementação *opensource* para o Ginga, denominada OpenGinga, que se adequa perfeitamente à resolução do nosso problema e será utilizada para validação dos nossos estudos de caso. Além de tudo, o ambiente proposto contribuirá com a validação dos componentes para o *middleware* Ginga e com a proliferação entre a comunidade de aplicações interativas baseadas no mesmo.

1.2 TRABALHOS CORRELATOS NA INSTITUIÇÃO

Desde sua fundação em 2003, o LAViD coordena e participa de diversos projetos de pesquisa e desenvolvimento na área de TVDI, sendo alguns deles importantes na contextualização deste trabalho. A seguir, um resumo dos principais projetos correlatos ao desenvolvimento do Xtation:

- FlexTV: Projeto realizado entre os anos de 2005 e 2006, e que teve como principais objetivos a concepção de uma arquitetura conceitual de um *middleware* e a definição de bibliotecas para programas de televisão interativos utilizadas como parte do Modelo de Referência do Sistema Brasileiro de Televisão Digital (SBTVD). A validação deste projeto foi realizada

a partir da implementação de um protótipo de *middleware*, que posteriormente serviu de base para a especificação do Ginga.

- Middleware Ginga: Projeto iniciado desde 2006 em parceria com a Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) e que tem como principal objetivo o desenvolvimento de uma implementação de referência do *middleware* Ginga juntamente com a elaboração de especificações utilizadas para a realização de testes de conformidade dos *middlewares* desenvolvidos para os equipamentos de TVDI. Dentro deste projeto, o LAViD vem trabalhando em uma implementação *opensource* do subsistema procedural do Ginga, denominado OpenGinga. Maiores detalhes sobre o Ginga e o OpenGinga são abordados no Capítulo 2.
- APLGinga – Avaliação do Middleware Ginga: Projeto realizado entre os anos de 2007 a 2009 em parceria com outras universidades e com o Centro de Pesquisas Renato Archer (CenPRA), com o objetivo de disseminar o desenvolvimento de aplicações e serviços para o Ginga, através da utilização de ferramentas gráficas que facilitem a autoria de aplicações interativas. Além disso, o projeto previa a avaliação do *middleware* Ginga por meio de ferramentas que propiciem o teste de aplicações, de modo que alguns componentes utilizados neste trabalho (ver item 4.2) foram desenvolvidos e/ou aprimorados durante este projeto.
- GingaCDN - Ginga Code Development Network: Projeto que faz parte do programa Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC), atualmente incubado pela Rede Nacional de Ensino e Pesquisa (RNP), e que tem como participantes diversas universidades brasileiras, dentre elas a UFPB. Iniciado em 2009, prevê a criação de uma rede de desenvolvedores de componentes e aplicações para o *middleware* brasileiro de TV Digital, permitindo o desenvolvimento distribuído e colaborativo da implementação de *middleware* OpenGinga.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O principal objetivo do trabalho é o desenvolvimento e validação de um ambiente que permita a execução e teste de aplicações para TVDI, através da utilização de ferramentas de transmissão, recepção e exibição de fluxos de áudio, vídeo e dados (aplicações) em uma TV acoplada ao ambiente, e de uma ferramenta que controle todo o processo de execução e testes.

1.3.2 Objetivos Específicos

Para que o objetivo geral do trabalho fosse atingido, foram definidos os seguintes objetivos específicos:

- Estudo do ambiente de TVDI, de seus componentes e das tecnologias relacionadas;
- Estudo dos conceitos relacionados às aplicações e programas interativos e ao seu processo de produção, com foco na fase de testes;
- Análise das ferramentas semelhantes disponíveis no mercado e de suas limitações;
- Definição de uma arquitetura distribuída que modele um sistema de TVDI de forma confiável e de baixo custo;
- Análise, modelagem, implementação e validação do ambiente proposto;
- Desenvolvimento de estudos de caso e avaliação.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está estruturada em seis capítulos. Seguindo esta introdução, tem-se no Capítulo 2 a fundamentação teórica relevante para este trabalho, onde são detalhados alguns conceitos relacionados à TVDI e seus componentes, à arquitetura do Ginga, às aplicações e programas interativos e ao seu processo de produção.

No Capítulo 3, são apresentados alguns trabalhos cujas características são similares ao ambiente proposto; sendo levantadas as suas principais características e limitações. No Capítulo 4 é apresentado o Xtation, sua arquitetura e aplicabilidade, e todo o processo de análise, projeto, modelagem e desenvolvimento do mesmo.

O Capítulo 5 traz estudos de caso que, baseados nos possíveis cenários de utilização do Xtation, tem o objetivo de validar o trabalho proposto. Por fim, as considerações finais são dispostas no Capítulo 6.

2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o contexto teórico das principais áreas temáticas que fundamentam este trabalho: a TV Digital Interativa e seus componentes; a especificação de *middleware* Ginga e a implementação OpenGinga; os conceitos e as características das aplicações e dos programas interativos; e o processo de produção de conteúdo para esta nova mídia, cujas fases exemplificam a aplicabilidade do trabalho ora proposto.

2.1. A TV DIGITAL INTERATIVA

A TV Digital Interativa é uma evolução da TV convencional, associando a esta sinais digitais e interatividade (RIBEIRO, 2004). Em outras palavras, a TVDI pode ser considerada a fusão entre a TV Digital (TVD) e a TV Interativa (TVi).

Um grande problema encontrado ao tentarmos descrever a TVDI está na abundância dos diferentes conceitos e entendimentos atualmente em circulação (JENSEN, 2007). Ainda que haja uma interpretação abrangente do que se entende por TV Interativa, ela não deve ser confundida com TV Digital, pois, apesar de haver uma ampla interseção entre estes dois conceitos, eles são, em si, diferentes (MÉDOLA, et al., 2007).

Apesar de existir há mais de 50 anos², não existe uma definição clara do que é a TV Interativa, sendo que algumas vezes os conceitos confundem-se com o de TVDI ou surge uma interpretação errônea de que a TVi nada mais é do que Internet na televisão, sem qualquer relação com o conteúdo audiovisual (SOUTO MAIOR, 2002).

A Tabela 1 lista algumas das principais definições de TVi encontradas na literatura.

Tabela 1: Definições de TV Interativa encontradas na literatura

<p>Mark Gawlinski (GAWLINSKI, 2003 p. 5)</p>	<p>[...] algo que permite que o telespectador ou telespectadores e as pessoas que fazem um canal de televisão, programa ou serviço se engajem em um diálogo. Mais especificamente, pode ser definida como um diálogo que leva os telespectadores a além da experiência passiva de assistir e os permita fazer escolhas e tomar ações.</p>
<p>Konstantinos Chorianopoulos (CHORIANOPOULOS, 2004 p. 9)</p>	<p>[...] é um termo genérico utilizado para todos os sistemas de televisão que oferecem ao consumidor interatividade além da troca de canais e do teletexto.</p>
<p>Jerry Whitaker (WHITAKER, 2001)</p>	<p>[...] tudo o que permite que um consumidor interaja com o sistema usando um controle remoto ou um teclado para acessar novos e avançados serviços.</p>
<p>Robert Schwalb (SCHWALB, 2003)</p>	<p>[...] é a coleção de serviços que suporta escolhas e ações iniciadas pelo telespectador e que são relacionadas a um ou mais canais de programação de vídeo.</p>
<p>Karyn Y. Lu (LU, 2005)</p>	<p>[...] qualquer programa de televisão que incorpora conteúdo adicional ou algum tipo de interatividade com o usuário.</p> <p>[...] termo genérico que cobre a convergência da televisão com tecnologias das mídias digitais como computadores, PVRs, jogos eletrônicos, dispositivos móveis e sem fio, possibilitando a interação com o usuário.</p>

² Nos anos 50, uma forma de TV interativa simples, mas inteligente foi criada na série infantil *Winky Dink and You* (1953-1957) da rede CBS. A interação era feita através de uma folha de plástico especial que as crianças colocavam na tela da televisão e sobre a qual elas desenhavam com um giz especial. Apesar de a tecnologia ser rudimentar, as crianças tinham uma experiência de interação com a televisão e podiam ver ações na tela que (aparentemente) eram em resposta aos seus desenhos (CAREY, 1996).

Diante desta diversidade de termos, fica-nos a idéia de que a principal intenção da TV Interativa é transformar-se em um meio mais interessante do que a televisão convencional no que se refere à recepção, à percepção e ao relacionamento do indivíduo com o meio, produzindo novos comportamentos do telespectador diante da tela através da possibilidade de controle sobre o que, quando e como assistir a um determinado conteúdo³.

Além disto, a interatividade na TV tornou-se um grande atrativo para a indústria da radiodifusão que está sempre em busca de novas maneiras de manter os telespectadores assistindo os seus produtos e, por conseqüência, suscitar novas oportunidades de gerar recursos, como nos sugere Steven Morris (MORRIS, et al., 2005).

Segundo (MONTEZ, et al., 2005), a televisão interativa é uma nova mídia porque quebra os paradigmas da televisão analógica de unidirecionalidade da informação e da passividade do telespectador. Outro ponto importante a ser considerado é que o modelo de negócio que sustenta a televisão passa a se basear no comércio, de forma que “o mercado, através de suas agências, também passa a atuar diretamente na televisão não apenas em propagandas, mas em serviços interativos através de aplicativos que permitem a venda ou o relacionamento direto com o usuário” (MÉDOLA, et al., 2007 p. 5).

As principais características e divergências entre a TV tradicional e a TV interativa estão listadas na Figura 1. Todas estas mudanças têm reflexo direto inclusive nas atitudes de quem assiste televisão. “Para os telespectadores, acostumados a simplicidade da televisão analógica, não há dúvida de que a introdução desses novos paradigmas demandará uma nova postura” (BECKER, 2006 p. 39).

³ Conteúdo é conjunto de informações contido em um programa ou programação. Refere-se também à natureza ou às características sociais e culturais destas informações (ANATEL – Agência Nacional de Telecomunicações, 2001). Na TV Interativa, o conteúdo deixa de ser apenas ditado por seu gerador e passa a incorporar caminhos determinados pelo telespectador.

Compreensão da televisão tradicional <i>Assumptions of Traditional Television</i>	Compreensão da televisão interativa <i>Assumptions of Interactive Television</i>
	
Expectador passivo <i>Passive Viewer</i>	Participante ativo (usuário) <i>Active Participant</i>
O conteúdo é empurrado até a audiência <i>Push Model</i>	O conteúdo é puxado pela audiência <i>Pull Model</i>
Modelo de receita baseado na propaganda <i>Advertising Driven Revenue Model</i>	Modelo de receita baseado no comércio <i>Commerce Driven Revenue Model</i>
Modelo de programação em função dos canais de transmissão <i>Broadcast Programming Model</i>	Modelo de programação com conteúdo disponível em banco de dados <i>Library Programming Model</i>
Programação linear <i>Linear Programming</i>	Programação participativa <i>Participatory Programming</i>
Dispositivo Central <i>Device Centric</i>	Ubiquidade (em qualquer lugar) <i>Ubiquitous</i>
Uso principal para o entretenimento <i>Usage Primarily for Entertainment</i>	Uso estendido a compras, comunicação, integração social e educação. <i>Usage Expands to Include Shopping, Communications, Social Integrations, and Education</i>
Plataforma de comunicação de uma só direção <i>One-way Communications Platform</i>	Plataforma de comunicação bidirecional <i>Two-way Communications Platform</i>
Os emissores conhecem bem o seu papel <i>Role of the Players is Well Understood</i>	O papel do emissor passa a ser flexível e requer maior integração <i>Role of the Players is Fluid and Requires Significant Integration</i>

Figura 1: Comparativo entre TV tradicional e TV Interativa. Fonte: (ADAMS, et al., 2001) apud (MÉDOLA, et al., 2007 p. 4).

Apesar do conceito de interatividade já ser explorado na televisão analógica, é a digitalização do sinal a responsável por ampliar, consideravelmente, o potencial dos serviços interativos na televisão. Ao aliarmos os conceitos de TV Interativa aos de TV Digital, temos à nossa disposição uma nova mídia que traz consigo uma miríade de possibilidades e expectativas.

A transmissão digital de televisão, grosso modo, implica transformar o sinal analógico da TV em informação digital. Como a transmissão é feita através de um fluxo de bits comprimido, há a possibilidade de se transmitir uma maior quantidade de informação, em comparação ao sistema analógico. A banda de 6MHz (largura de banda utilizada para a transmissão de canais de TV no Brasil) que era usada para transportar apenas 1 canal de vídeo analógico, pode agora ser usada para transmitir de 4 a 10 canais digitais.

Entretanto, o grande diferencial da transmissão digital não está apenas na quantidade de canais que ela permite transmitir; está na possibilidade de transmitir dados juntamente ao sinal de vídeo (SOUTO MAIOR, 2002). O telespectador passa, então, a receber sinais de áudio e vídeo associados a dados fornecidos por meio de aplicações (*software*) a partir das operadoras de TV. Alta definição de imagem e som (HDTV – *High Definition Television*), mobilidade e portabilidade são outros grandes atrativos desta nova tecnologia.

No Sistema Brasileiro de Televisão Digital Terrestre (SBTVD-T)⁴ todas as características acima listadas foram contempladas, conforme sancionado pelo Presidente da República através do Decreto nº 5.820, de 29 de junho de 2006:

Art. 6º O SBTVD-T possibilitará:

- I - transmissão digital em alta definição (HDTV) e em definição padrão (SDTV);
- II - transmissão digital simultânea para recepção fixa, móvel e portátil; e
- III - interatividade.

A fim de garantir que todas estas novas funcionalidades cheguem às televisões e aos receptores móveis, encontra-se detalhada a seguir a infraestrutura necessária ao funcionamento da TVDI.

2.1.1 COMPONENTES DA TV DIGITAL INTERATIVA

Um sistema de TVDI pode ser decomposto em três partes principais, conforme ilustrado na Figura 2:

- (i) um difusor, responsável por prover o conteúdo a ser transmitido e dar suporte às interações dos telespectadores;
- (ii) um meio de difusão, que habilita a comunicação entre o difusor e o receptor; e
- (iii) um receptor, que recebe o conteúdo e oferece a possibilidade do telespectador reagir ou interagir com o difusor.

⁴ SBTVD-T é o conjunto de padrões tecnológicos a serem adotados para transmissão e recepção de sinais digitais terrestres de radiodifusão de sons e imagens (BRASIL, 2006)

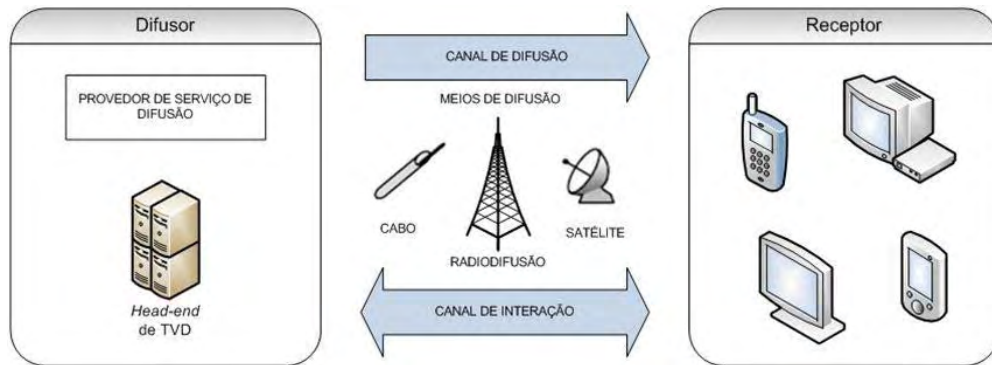


Figura 2: Modelo de um sistema de TVDI. Fonte: (BATISTA, 2007), com adaptações.

A difusão é o envio de conteúdo (áudio, vídeo e dados) de um ponto provedor do serviço, responsável pelo gerenciamento de diversos canais televisivos, para os receptores. A construção do sinal a ser transmitido implica em uma série de etapas, como mostra a Figura 3.

Um conteúdo televisivo pode ser gerado basicamente de duas formas: transmitindo-o ao vivo ou gravando vários fluxos de vídeo e áudio para serem editados antes da difusão. Em ambas as formas, a aquisição dos sinais de vídeo e áudio pode ser através de câmeras filmadoras digitais ou analógicas. No entanto, para poderem ser transmitidos na forma digital, os sinais de áudio e vídeo precisam ser codificados. O codificador de vídeo é o equipamento responsável por codificar uma entrada de vídeo em fluxo de vídeo digital, de acordo com um determinado padrão de codificação (MPEG-2 ou H.264, por exemplo).

O processo de encapsulamento e de difusão de dados (*data broadcasting*) juntamente com o fluxo de vídeo/áudio digital é conhecido como *datacasting*. Na TVDI, o *datacasting* é realizado através do Gerador de Carrossel, módulo responsável por gerar um fluxo elementar de um sistema de arquivos (comumente aplicações interativas e seus arquivos) de maneira cíclica, produzindo o carrossel de dados. Com o envio periódico de dados, o receptor/telespectador apenas aguarda o próximo envio quando precisar de uma determinada informação adicional. A implementação do carrossel de dados é baseada no protocolo DSM-CC (*Digital Storage Media Command and Control Protocol*).

O Servidor de SI (*Service Information*) é o dispositivo responsável por gerenciar as tabelas que fornecem informações sobre o serviço oferecido em um canal de TV – tipicamente as informações acerca dos fluxos de áudio e vídeo transmitidos nesse canal.

As saídas resultantes dos componentes acima descritos servem de entrada para um equipamento chamado Multiplexador, responsável por unir (multiplexar) todos os fluxos que compõem o fluxo a ser transmitido. O formato adotado pela maioria dos sistemas é o MPEG-2 TS (*MPEG-2 Transport Stream*), que suporta multiplexação de múltiplas instâncias de vídeo, áudio e conjunto de dados (aplicações e seus arquivos). Após a multiplexação, ocorre a modulação do sinal digital, permitindo a sua difusão através dos meios de difusão.

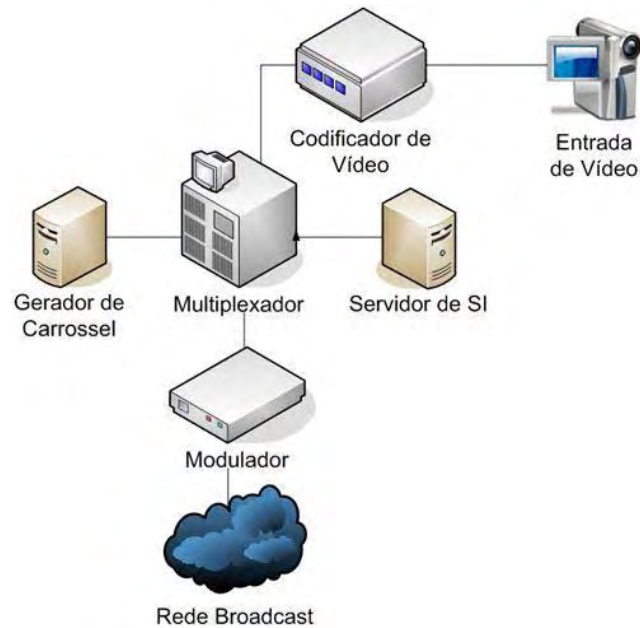


Figura 3: Etapas da difusão do conteúdo televisivo. Fonte: (BATISTA, 2007).

Os meios de difusão mais comuns são via satélite, cabo e radiodifusão, também conhecido como difusão terrestre. O SBTVD-T é um exemplo de sistema que utiliza a radiodifusão terrestre.

A fim de melhor explorar o potencial da interatividade, os meios de difusão também são utilizados para o canal de retorno (canal de interação), permitindo que os telespectadores possam enviar informações de volta à estação de TV utilizando-se, por exemplo, de uma conexão com a Internet. Várias tecnologias podem ser utilizadas como canal de interação, por exemplo, telefonia celular, telefonia fixa, rádio, satélite. Estas tecnologias possuem uma relação diretamente proporcional à complexidade dos serviços oferecidos. Ou seja, quanto maior a banda, maior pode ser a complexidade da interatividade. Ou quanto mais complexos os serviços interativos oferecidos, maior deve ser a banda disponível no canal (MONTEZ, et al., 2005).

Quando não há canal de retorno, o usuário recebe em seu aparelho receptor um pacote de aplicativos e pode dispor da escolha e formatação do conteúdo que consome por meio de alternativas que o emissor propõe.

O receptor pode estar embutido em uma televisão digital ou ser um equipamento a parte, o *Set-top Box* (STB), responsável por receber o sinal transmitido, decodificá-lo e exibi-lo. O STB pode ser visto como um computador adaptado às necessidades do ambiente televisivo, possuindo processador, memória, sistema operacional, etc. Para permitir ao telespectador a interação através de canal de retorno, o STB deve possuir também uma interface compatível com o canal de interação utilizado. Dispositivos móveis, como PDAs e celulares, também podem ser utilizados como receptores de TVDI.

Um receptor de TVDI possui uma arquitetura em camadas, conforme ilustrado na Figura 4. Através deste modelo de arquitetura, cada camada oferece serviços para a camada superior e usa os serviços oferecidos pela camada inferior.



Figura 4: Arquitetura em camadas de um receptor de TVDI

Desta forma, o sistema operacional trata de mecanismos como transmissão e sincronização de mídia, manipulando os fluxos fornecidos pelo *hardware* e fazendo a interface com a camada superior.

Já o *middleware* é a instância de *software* responsável por abstrair características específicas de cada receptor, permitindo que uma mesma aplicação seja executada em *Set-top Boxes* com fabricantes e especificações diferentes (SOUZA FILHO, et al., 2007).

Neste caso, garantir a portabilidade é primordial, tendo em vista que as tecnologias de TVDI permitem a fabricação de STBs com diferentes arquiteturas de *hardware*, cujas capacidades de processamento, armazenamento e comunicação são bastante variáveis.

Em um cenário de *hardware* e *software* heterogêneos, os desenvolvedores de aplicações deveriam escrever diferentes versões dos programas para cada combinação de *hardware* e sistema operacional dos diversos tipos de STBs. Esta heterogeneidade das plataformas tornaria o desenvolvimento de aplicações para TVDI uma atividade ineficiente e de custo elevado, que poderia inviabilizar sua adoção em larga escala. A fim de contornar esta situação, o *middleware* provê facilidades às aplicações na forma de interfaces de programação padronizadas (APIs, do inglês *Application Programming Interfaces*), tornando-as portáveis nos diversos tipos de STBs.

Os principais projetos de *middleware* utilizados mundialmente são o MHP (*Multimedia Home Platform*)⁵, do padrão europeu DVB (*Digital Video Broadcast*)⁶; os americanos DASE (*Digital TV Applications Software Environment*)⁷, OCAP (*OpenCable Application Platform*)⁸ e ACAP (*Advanced Common Application Platform*)⁹, utilizados no sistema ATSC (*Advanced Television Systems Committee*)¹⁰, e o japonês ARIB (*Association of Radio Industries and Businesses*)¹¹, do sistema ISDB (*Integrated Services Digital Broadcasting*)¹². Recentemente o Brasil decidiu por especificar um *middleware* próprio, o Ginga.

Diante da grande importância desta camada para o desenvolvimento das aplicações, na seção seguinte iremos abordar em mais detalhes a especificação e a implementação de *middleware* utilizadas como base deste trabalho, o Ginga e o OpenGinga.

⁵ <http://www.mhp.org/>

⁶ <http://www.dvb.org/>

⁷ <http://www.atsc.org/cms/index.php/standards/published-standards/71-atsc-a100-standard>

⁸ <http://www.cablelabs.com/opencable/>

⁹ http://www.atsc.org/cms/standards/a_101a.pdf

¹⁰ <http://www.atsc.org/cms/>

¹¹ <http://www.arib.or.jp/english/>

¹² <http://www.dibeg.org/>

2.2 O MIDDLEWARE GINGA E A IMPLEMENTAÇÃO OPENGINGA

O Ginga é a especificação de *middleware* do Sistema Brasileiro de Televisão Digital, resultado da fusão dos *middlewares* FlexTV e MAESTRO, desenvolvidos por consórcios liderados pela UFPB e PUC-Rio, respectivamente.

O FlexTV (LEITE, et al., 2005), *middleware* procedural de referência do SBTVD, incluía um conjunto de APIs estabelecidas internacionalmente e funcionalidades inovadoras, como a possibilidade de gerenciamento de conexões com múltiplos dispositivos, permitindo que os mesmos fossem utilizados para interagir simultaneamente com uma única aplicação.

Já o MAESTRO (SOARES, 2006), o *middleware* declarativo, era responsável por processar documentos escritos na linguagem declarativa NCL¹³ (*Nested Context Language*) e na linguagem de script Lua¹⁴, tendo como principal facilidade o sincronismo espaço-temporal entre objetos multimídia.

O *middleware* Ginga integrou estas duas soluções, agora chamadas de Ginga-J (SOUZA FILHO, et al., 2007) e Ginga-NCL (SOARES, et al., 2007), padronizadas nas recomendações internacionais da *International Telecommunication Union* - ITU: J.200 (ITU, 2001), J.201 (ITU, 2004) e J.202 (ITU, 2003).

Desta forma, o *middleware* Ginga é subdividido em dois subsistemas principais interligados (Figura 5), também chamados de Máquina de Execução e Máquina de Apresentação, sendo um para aplicações procedurais Java e outro para aplicações declarativas NCL. A execução do conteúdo procedural torna-se possível através da Máquina Virtual Java (JVM, do inglês *Java Virtual Machine*). Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma de programação será mais adequado que o outro.

Os dois ambientes para execução de aplicações interativas não são necessariamente independentes, uma vez que a recomendação do ITU inclui uma “ponte” que deve disponibilizar APIs para intercomunicação entre os mesmos. Essa API de ponte permite que as aplicações declarativas utilizem serviços disponíveis nas aplicações procedurais, e vice-

¹³ <http://www.ncl.org.br>

¹⁴ <http://www.lua.org>

versa. Considerando que as aplicações são executadas na camada acima da camada dos ambientes de execução, é possível que as aplicações possuam uma arquitetura híbrida, com partes declarativas e procedurais.

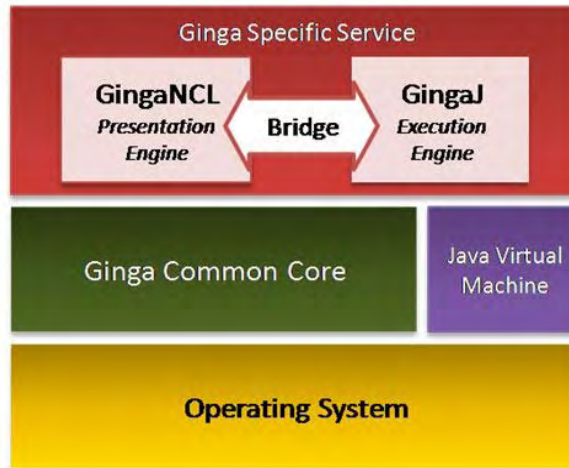


Figura 5: Arquitetura em alto nível do middleware Ginga. Fonte: (SOUZA FILHO, et al., 2007)

O *Ginga Common Core* (Ginga-CC) é o subsistema do Ginga responsável por oferecer serviços de infraestrutura de TV Digital, abstraindo o sistema operacional e o *hardware* para as outras camadas e oferecendo o suporte necessário aos ambientes declarativo e procedural. Como suas principais funções, podemos citar a exibição dos vários objetos de mídia, o controle do plano gráfico, o tratamento do canal de retorno e a sintonização dos diferentes tipos de canais.

A definição Ginga-J é composta por um conjunto de APIs projetadas para suprir todas as funcionalidades necessárias para a implementação de aplicativos para televisão digital, desde a manipulação de dados multimídia até protocolos de acesso. Sua especificação é formada por uma extensão à API do Serviço de Informação ISDB ARIB B.23, pela especificação Java DTV e por um conjunto de APIs adicionais (Figura 6).



Figura 6: Conjunto de APIs que compõe a especificação do Ginga-J. Fonte: (SUN, 2008)

As APIs adicionais incluem o conjunto de classes disponíveis para a ponte entre os aplicativos escritos nas linguagens NCL e Java, uma especificação de API para o canal de interatividade e um conjunto de funcionalidades inovadoras para a integração de dispositivos, viabilizando o suporte a recursos multimídia e interação simultânea de múltiplos usuários em aplicações de TVDI (SILVA, et al., 2007) (SILVA, 2008).

A especificação Java DTV (SUN, 2008) é uma plataforma aberta, interoperável e sem cobrança de *royalties* que permite a implementação de serviços interativos com a linguagem Java, tendo sido inserida recentemente ao conjunto de APIs do Ginga-J. Funcionalmente, a JavaDTV substitui a coleção de APIs utilizada anteriormente e definidas pelo padrão GEM (*Globally Executable MHP*), tais como DAVIC (*Digital Audio Video Council*) e HAVi (*Home Audio Video Interoperability*), com o intuito de prover uma solução livre de *royalties* para os fabricantes de dispositivos e desenvolvedores de aplicações e permitir a produção de aparelhos de TV e/ou conversores por um custo mais acessível.

Esta especificação é composta pelas APIs Java DTV e JavaTV¹⁵, adicionadas à base de componentes do ambiente de execução Java (*Java Runtime*), incluindo as APIs *Connected Device Configuration*, *Foundation Profile* e *Personal Basis Profile* (Figura 7). Dentre as principais novidades da Java DTV quanto ao desenvolvimento de aplicações, podemos citar a API LWUIT¹⁶ (*LightWeight User Interface Toolkit*), responsável por definir os componentes gráficos e o mecanismo de tratamento de eventos de usuário.

¹⁵ <http://java.sun.com/javame/technology/javatv/>

¹⁶ http://java.sun.com/developer/technicalArticles/javame/lwuit_intro/

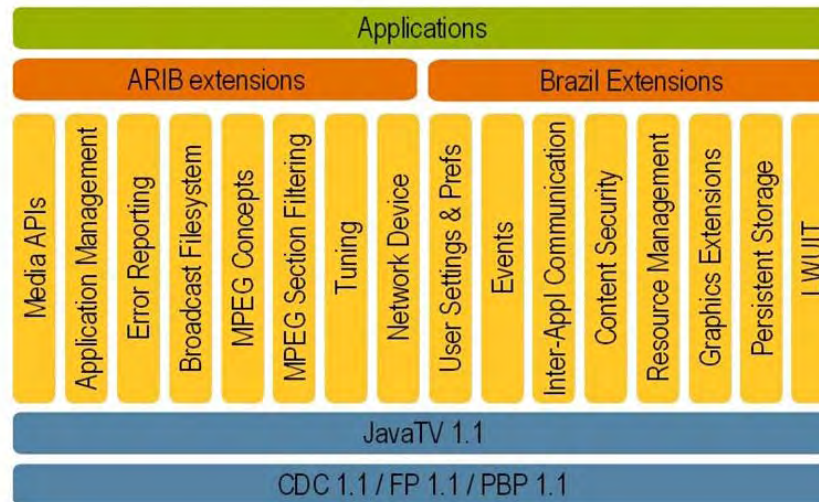


Figura 7: Conjunto de APIs Java DTV. Fonte: (SUN, 2008).

Como forma de validação da especificação do *middleware* de referência do SBTVD, a UFPB, através do LAVID, e a PUC-Rio vêm trabalhando em uma implementação de referência do Ginga, o OpenGinga¹⁷.

O OpenGinga é um projeto de código aberto com licença GNU GPL (*General Public License*), que funciona como um *Set-top box* virtual, capaz de sintonizar fluxos de vídeo e executar aplicações Ginga em um computador pessoal. A princípio, a versão disponível do OpenGinga possui suporte apenas à execução de aplicações Java. Sua arquitetura é totalmente baseada na do Ginga-J.

Como forma de impulsionar o desenvolvimento colaborativo do OpenGinga e de abrir um acesso livre às tecnologias relacionadas ao *middleware* do SBTVD, ampliando, por consequência, a competência nacional para inovações nesta área, foi criada uma rede de desenvolvedores de componentes e ferramentas para o *middleware* brasileiro, o *Ginga Code Development Network* ou simplesmente GingaCDN¹⁸.

O GingaCDN tem uma filosofia de desenvolvimento distribuído e colaborativo de *software*, de forma que os interessados podem trabalhar no desenvolvimento de componentes e submetê-los à rede. Um ponto importante a ser observado diz respeito à necessidade de provimento de uma forma de validação destes componentes e de realização de testes de conformidade e integração das implementações propostas para o *middleware*.

¹⁷ <http://www.openginga.org/>

¹⁸ <http://gingacdn.lavid.ufpb.br/index.html>

2.3 APLICAÇÕES E PROGRAMAS INTERATIVOS PARA TVDI

As mudanças da TV trazidas pela implantação da televisão digital são significativas, implicando uma nova maneira de produzir e realizar programas televisivos (BECKER, 2006). Isso se dá principalmente pelo acréscimo de um novo componente ao processo: o *software*.

O *software* transmitido juntamente com o conteúdo audiovisual é chamado, na TV Digital Interativa, de aplicativo ou aplicação. No entanto, as aplicações de TVDI têm algumas particularidades em relação às tradicionais aplicações de *software* (VEIGA, et al., 2007):

- (i) Podem fazer parte de um programa de TV convencional (aqui denominados apenas de programa de TV) o qual tem formato e contexto próprios;
- (ii) Apesar de serem oferecidas por meio de um *front-end* único (a TV) podem atender a mais de um usuário. Geralmente a TV atende uma família onde cada telespectador tem um perfil particular, cujos anseios em um veículo de comunicação de massa são mapeados em diversidade na grade de programação;
- (iii) Devem lidar com interatividade (em diferentes níveis) e organização não-linear de conteúdo;
- (iv) Exigem uma infraestrutura de transmissão e componentes de *software* e *hardware* adequados.

Quanto à relação da aplicação interativa com o conteúdo audiovisual do programa de TV, Becker (BECKER, 2006) apresenta uma classificação que define:

- aplicativo: todo software de TVDI;
- aplicativo desacoplado: tipo de aplicativo que não tem relação com conteúdo audiovisual do programa de TV;
- aplicativo acoplado: tipo de aplicativo que tem relação com o conteúdo audiovisual do programa.

Cabe aqui ressaltar que os aplicativos possuem tempo de vida próprio, que pode estar ou não relacionado ao tempo de vida de um programa de TV. Estes aplicativos funcionam como uma máquina de estados; e em seu ciclo de vida podem estar carregados,

pausados, em execução ou destruídos, como mostra a Figura 8. As transições entre estes estados podem ser ativadas por eventos disparados pela própria aplicação, pelo conteúdo audiovisual relacionado, pelo usuário ou pelo próprio ambiente de execução.

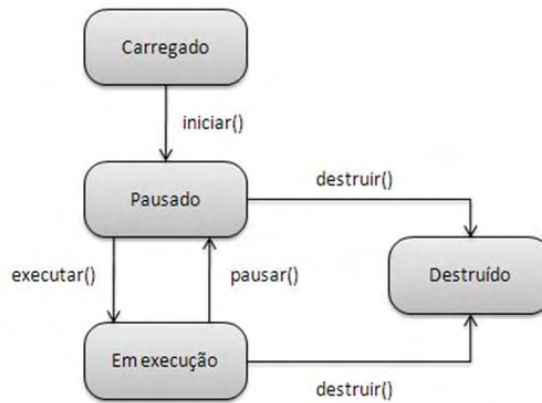


Figura 8: Ciclo de vida das aplicações de TVDI. Fonte: (MORRIS, et al., 2005), tradução nossa.

Exemplos típicos de aplicativos desacoplados são envio de mensagens (SMS, correio eletrônico), jogos e o T-banking, ilustrados na Figura 9. Estas aplicações são independentes do conteúdo audiovisual e não precisam especificamente de um vínculo com um programa de TV. Neste caso, a televisão é utilizada apenas como mais um meio de acesso ao serviço; outros meios poderiam ser utilizados, como o celular ou a *web*. O foco do usuário está na atividade em si e, por esta razão, (BARROS, 2006) denomina esta classe de aplicativos como TV Ativa.



Figura 9: Exemplos de aplicativos sem relação com conteúdo audiovisual do programa de TV (a) correio eletrônico; (b) envio de SMS; (c) jogo; (d) T-banking. Fonte: www.broadbandbananas.com

As aplicações acopladas, por sua vez, estão relacionadas de alguma forma ao conteúdo audiovisual de um programa de TV e ao seu tempo de vida. Portanto, elas podem estar disponíveis apenas em algum momento do programa de TV, durante todo o seu tempo de exibição, ou até mesmo além deste tempo.

Em alguns casos, os aplicativos acoplados possuem um momento exato para serem disparados e outro para serem abortados, de acordo com determinados eventos relacionados ao vídeo. São também denominados de fortemente acoplados (BECKER, 2006). Como exemplo, temos um programa de TV que lança uma aplicação de enquete ao vivo sobre um determinado assunto (Figura 10). Após o assunto se esgotar ou se houver alteração no tema abordado, a aplicação deverá sair do ar.



Figura 10: Protótipo de programa de TVDI com aplicação fortemente acoplada. Durante um determinado momento na transmissão do carnaval carioca, o telespectador pode escolher a musa do carnaval utilizando o controle remoto para participar de uma enquete ao vivo. Fonte: acervo do LAVID.

Uma aplicação pode também estar relacionada ao vídeo, mas sem a obrigatoriedade de ser executada em uma ocasião específica, podendo iniciar e morrer em um momento a escolha do usuário, durante o tempo de vida do programa de TV. Por exemplo, o usuário pode ter acesso a informações adicionais às reportagens de um jornal, enquanto o programa estiver no ar (Figura 11), visualizar informações extras sobre uma partida de futebol, durante a transmissão do jogo (Figura 12), ou ainda, acompanhar em tempo real o resultado das eleições, durante a transmissão do programa relacionado (Figura 13).



Figura 11: Exemplo aplicação para a categoria de notícias, na qual o telespectador tem acesso a recursos extras (vídeos e informações), enquanto assiste o jornal. Fonte: www.broadbandbananas.com



Figura 12: Exemplo de aplicação na categoria esportiva, onde o usuário pode acessar diversas informações extras sobre a partida de futebol durante sua exibição. Fonte: www.broadbandbananas.com



Figura 13: Exemplo de aplicação para as eleições. Os telespectadores podem acompanhar em tempo real o resultado da votação em diferentes partes do Reino Unido, durante a exibição do programa “itv Election 2001”. Fonte: (GAWLINSKI, 2003)

Há ainda os aplicativos que, apesar de também estarem relacionados a algum conteúdo audiovisual, não tem momento definido para sua execução, ficando acessíveis mesmo após o término do programa de TV. Por exemplo, resumos de programas que ficam no ar durante a semana para quem não os assistiu; ou enquetes sobre temas relacionados a um programa que podem coletar respostas durante a semana (BECKER, 2006). Outro exemplo é uma aplicação para arrecadação de doações, como a ilustrada na Figura 14. A aplicação pode ficar disponível durante todo o período da campanha, permitindo assim a maior participação dos telespectadores.



Figura 14: Exemplo de aplicativo sem momento definido para execução. Através da aplicação relacionada ao programa “BBC Children in Need Telethon”, os telespectadores escolhem a quantia que desejam doar à campanha e inserem os dados de seu cartão de crédito, usando o controle remoto.

Fonte: www.broadbandbananas.com

Nesta classe de aplicativos acoplados, também conhecida na literatura como *Enhanced TV* (GAWLINSKI, 2003) (LAMONT, 2003) ou *TV Expandida (TVe)*, ou ainda *TV Avançada*, o usuário mantém o foco de atenção no programa em questão e conteúdo interativo existe em função do conteúdo audiovisual. Como este tipo de aplicativo está sempre associado a um programa de TV, pode ser chamado de programa de *TVe*, fazendo referência ao programa como um todo (incluindo aplicativo e conteúdo audiovisual) (BARROS, 2006 p. 19). Neste trabalho, daremos a ele também a denominação de programa interativo para *TVDI*, ou simplesmente programa de *TVDI*, nos referindo justamente à junção de um programa de TV a um *software* interativo.

Os programas de *TVDI* englobam textos, vídeos e diversos elementos gráficos, como imagens e animações, que têm por finalidade fornecer informações adicionais ou possibilitarem diversas formas de interação com o usuário. Encontramos exemplos de programas de *TVDI* nas mais diversas categorias, como esportes, notícias, entretenimento, etc. Um ponto interessante relacionado à interface visual destes programas, diz respeito à forma de exibição do vídeo em relação à aplicação interativa (TEIXEIRA, et al., 2008). Em alguns casos, o vídeo pode ser redimensionado de maneira a acomodar melhor a interface do aplicativo (Figura 13). Já em outros exemplos a interface da aplicação é carregada sobre o vídeo (*overlay*), utilizando-se muitas vezes de recursos de transparência para que o usuário não perca os detalhes do programa em exibição (Figura 12). Há ainda casos em que o vídeo principal dá lugar à interface da aplicação, que passa a ocupar toda a tela.

Outra forma de classificar as aplicações interativas diz respeito à linguagem de programação utilizada em seu desenvolvimento. De acordo com o paradigma utilizado, as aplicações pra TVDI podem ser classificadas em declarativas, imperativas ou híbridas (CPqD, 2006) (SOARES, et al., 2008).

Uma linguagem declarativa enfatiza a descrição declarativa do problema, ao invés da sua decomposição em uma implementação algorítmica. Elas são linguagens de mais alto nível de abstração, usualmente ligadas a um domínio ou objetivo específico, onde o programador fornece apenas o conjunto das tarefas a serem realizadas, não estando preocupado com os detalhes de como elas serão executadas (SOARES, 2007). A linguagem declarativa utilizada no contexto do SBTVD é a NCL. Um exemplo de aplicação desenvolvida nesta linguagem é ilustrado na Figura 15.

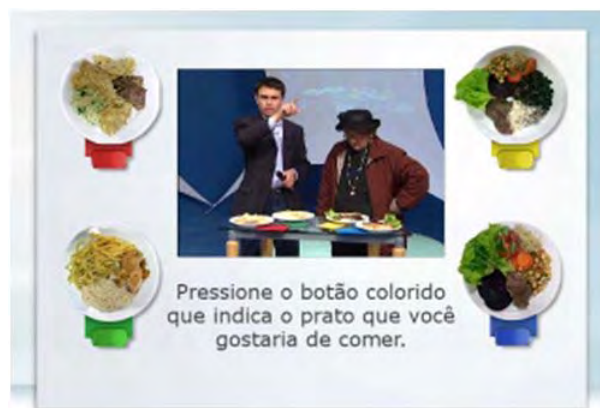


Figura 15: Exemplo de aplicação desenvolvida em NCL, através da qual o usuário pode interagir com o programa de TVDI “Viva Mais” e receber dicas sobre alimentação saudável. Fonte: <http://clube.ncl.org.br>

Já na programação imperativa (ou procedural), o computador deve ser informado sobre cada passo a ser executado. Pode-se afirmar que, em linguagens imperativas, o programador possui um maior poder sobre o código, sendo capaz de estabelecer todo o fluxo de controle e execução de seu programa. Entretanto, para isso, ele deve ser bem qualificado e conhecer bem os recursos de implementação. A linguagem mais usual encontrada nos ambientes imperativos de um sistema de TV digital é a Java. Uma aplicação para TVDI desenvolvida nesta linguagem é também conhecida como *Xlet*. A Figura 16 ilustra um programa de TVDI com aplicação desenvolvida na linguagem Java.



Figura 16: Protótipo de programa de TVDI para um jornal interativo, cuja aplicação foi desenvolvida na linguagem Java. Através do menu principal o usuário pode enviar sugestões, participar de enquetes, visualizar matérias e vídeos relacionados ao assunto principal do jornal, etc. Fonte: (ALVES, et al., 2009).

No entanto, uma aplicação não precisa ser puramente declarativa ou puramente imperativa. Frequentemente, aplicações declarativas fazem uso de conteúdos em *script* que são de natureza imperativa, e podem também referenciar um código imperativo embutido. Da mesma forma, aplicações imperativas podem referenciar conteúdos declarativos, ou até construir e iniciar a apresentação de um conteúdo declarativo. Estas aplicações são denominadas de híbridas constituem uma poderosa ferramenta de produção de conteúdo para TVDI, ao unir as vantagens dos paradigmas imperativo e declarativo.

Sob o ponto de vista da forma de interação com o usuário e utilização do canal de retorno, as aplicações interativas podem ser classificadas em aplicações de interatividade local, de interatividade intermitente e de interatividade permanente (SOUZA, et al., 2005) (WAISMAN, 2006).

Nas aplicações de interatividade local, o usuário não consegue estabelecer uma comunicação com o emissor, pelo fato deste tipo de interatividade não possuir o canal de retorno ou de interação. Todos os dados são transmitidos no fluxo e quando o usuário solicita uma informação, ela já está previamente armazenada no STB. Entre as aplicações providas com interatividade local, encontram-se multicâmeras, extras vinculados ao programa, portal de informação e o Guia Eletrônico de Programação (EPG, do inglês *Electronic Program Guide*) (Figura 17-a).

Nas aplicações de interatividade Intermitente já há a adoção de um canal de retorno para estabelecer uma comunicação assíncrona do usuário com o serviço oferecido. Para este

tipo de aplicações, a comunicação entre o telespectador e o emissor de conteúdo se dá de forma unidirecional, ou seja, o telespectador apenas envia informações para o emissor, sem esperar nenhuma resposta a cerca das informações enviadas. Como a comunicação é assíncrona, as informações podem ser processadas posteriormente, sem prejuízo da aplicação e da informação sob o ponto de vista do usuário. Neste tipo de aplicação se enquadram o envio de votos de enquetes, pesquisas de opinião, jogos do tipo quiz (Figura 17-b), envio de mensagens SMS, entre outras.

Já nas aplicações de interatividade permanente, há um canal de retorno que estabelece comunicação síncrona do usuário com os serviços oferecidos e até mesmo com outros usuários. A comunicação é feita de forma dedicada e bi-direcional, e o usuário permanece em tempo integral conectado ao canal de retorno, podendo enviar dados ao servidor a qualquer momento e de forma instantânea. Como exemplo, temos as aplicações de acesso a caixa de e-mails, bate-papo, jogos online, *T-commerce* (Figura 17-c), *T-banking* e *T-learning*.



Figura 17: Exemplo de aplicação de interatividade local, intermitente e permanente, respectivamente: (a) EPG; (b) Quiz; (c) T-commerce. Fonte: www.broadbandbananas.com.

A Figura 18 mostra um resumo das características dos aplicativos e programas interativos apresentadas neste tópico. Categorizá-los é uma tarefa difícil e abrangente e na literatura ainda não existe um consenso em termos da taxonomia usada sobre os tipos de aplicações interativas (GAWLINSKI, 2003). Como diversos atributos podem ser levados em consideração, os autores acabam por misturar classificações baseadas na tecnologia utilizada, no tipo de uso, na atividade comercial e até no tipo de emissora de forma indistinta (BARROS, 2006), comprometendo a categorização. Porém, considerando apenas as

características apresentadas e realizando combinações entre elas, já podemos visualizar o quão rica é a produção de conteúdo para TVDI.



Figura 18: Resumo das principais características das aplicações e programas de TVDI apresentadas.

2.4 PROCESSO DE PRODUÇÃO DE PROGRAMAS INTERATIVOS PARA TVDI

Como vimos nos tópicos anteriores, a TVDI traz uma grande novidade em relação à TV tradicional: a possibilidade vincular uma gama aplicações de computador ao conteúdo televisivo transmitido. Isto traz impactos significativos não só na forma de assistir TV, mas também na forma de produção do seu conteúdo.

Sabemos que aplicações interativas nada mais são do que softwares; e como tal, precisam passar por um processo que garanta a execução consistente e a estruturação das atividades que compõem o trabalho da equipe de desenvolvimento, além da adequação e qualidade do produto final.

Diversos modelos de processo de software são encontrados na literatura (PRESSMAN, 2002) (SOMMERVILLE, 2004), cada qual exibindo seus pontos fortes e fracos, mas todos tendo uma série de fases genéricas em comum, dentre elas a fase de definição, a de desenvolvimento e a de manutenção (PRESSMAN, 2002). Durante a fase de definição são realizadas as tarefas de engenharia de sistemas ou de informação, planejamento e análise de requisitos. Na fase de desenvolvimento, três tarefas técnicas específicas devem sempre ocorrer: projeto do software, geração de código e testes. A fase de manutenção aplica

novamente os passos das fases de definição e desenvolvimento, mas o faz no contexto de modificação do software existente, aplicando a manutenção corretiva, adaptativa, perfectiva e preventiva.

O grande desafio não está na produção das aplicações interativas em si, mas na produção dos programas de TVDI, que agregam mecanismos de interatividade ao programa de TV convencional a partir da vinculação de softwares interativos. Por esta razão, o processo de desenvolvimento de programas para TVDI integra atividades inerentes ao processo de produção de TV e atividades do desenvolvimento de software, conforme podemos observar na Figura 19.

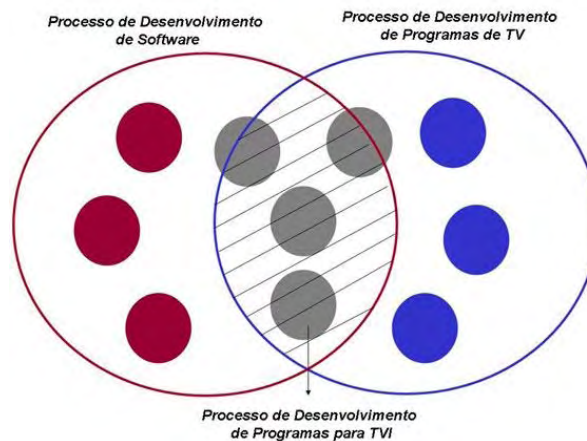


Figura 19: Processo de Desenvolvimento de Programas para TVDI. Fonte: (TAVARES, 2004)

O problema desta nova abordagem reside no seguinte fato: “a indústria de TV não tem cultura de desenvolver software, assim como a indústria de software não tem experiência em fazer programas para TV” (VEIGA, et al., 2007). Torna-se necessária uma metodologia que seja capaz de gerenciar tanto o processo que trata da produção e concepção do programa televisivo quanto o processo de implementação das aplicações interativas, através da colaboração de uma equipe multidisciplinar.

O processo de desenvolvimento de programas para TV convencional depende bastante da sistemática de trabalho da organização e do propósito do programa em desenvolvimento. Além disto, este processo vem sendo aperfeiçoado ao longo dos anos, de forma a tornar as tecnologias de produção mais fáceis de usar, tornar possível a execução de programas em todo o mundo e, principalmente, para facilitar as mudanças criativas em

qualquer ponto do processo (GAWLINSKI, 2003). “O objetivo desse constante aprendizado é o telespectador, que está do outro lado da tela, consumindo os produtos e os programas. Entender seus gostos e necessidades é um primeiro passo para alcançar o objetivo de uma produção” (BARBOSA NETO, 2008). A Tabela 2 sugere um modelo de processo de produção televisiva, com suas fases e tarefas relacionadas.

Tabela 2: Estágios da Produção Televisiva (GAWLINSKI, 2003).

	<i>Desenvolvimento</i>	<i>Pré-Produção</i>	<i>Produção</i>	<i>Pós-Produção</i>	<i>Broadcast</i>
<i>Exemplos do que é feito</i>	<ul style="list-style-type: none"> - BUSCA DE OPORTUNIDADES - DESENVOLVIMENTO DA IDÉIA - PATROCÍNIO - ESTUDOS SOBRE A IDÉIA 	<ul style="list-style-type: none"> - AGENDA DE PRODUÇÃO - RECRUTAMENTO - ORÇAMENTO - STORYBOARD - ROTEIRO - ESTUDOS SOBRE O PROGRAMA 	<ul style="list-style-type: none"> - GRAVAÇÃO 	<ul style="list-style-type: none"> - EDIÇÃO - EDIÇÃO GRÁFICA 	<ul style="list-style-type: none"> - EXIBIÇÃO - PROMOÇÃO E MARKETING

Tomando como base o desenvolvimento ágil de software (BECK, et al., 2001) (SOARES, 2004) e os modelos de produção de TV, (VEIGA, 2006) descreve em seu trabalho uma proposta de metodologia para o desenvolvimento de programas de TVDI.

Aproveitando as melhores práticas das metodologias ágeis XP (*eXtreme Programming*) e SCRUM, (VEIGA, 2006) sugere práticas desejáveis para uma metodologia que auxilie o desenvolvimento de programas para TVDI. São elas: (i) planejamento; (ii) ciclos curtos e iterativos; (iii) papéis multidisciplinares; (iv) trabalho colaborativo e interativo; (v) versões pequenas e funcionais; (vi) código simples; e (vi) modularidade.

Assim como acontece nos modelos de produção de programas de TV e de software, o ciclo de vida do desenvolvimento de programa de TVDI também pode ser dividido em fases, como mostra a Figura 20.

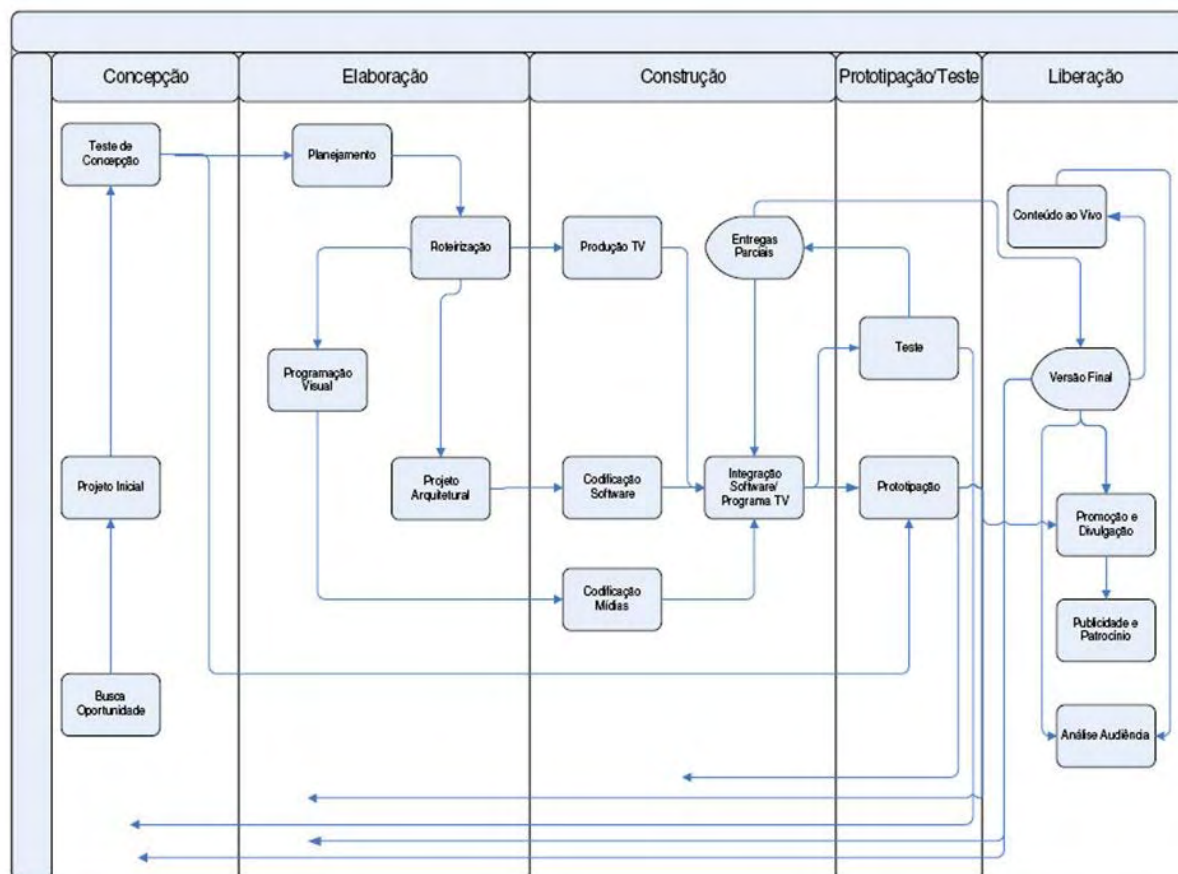


Figura 20: Fases do Modelo de Processo para Desenvolvimento de Programas de TVDI (VEIGA, 2006).

O processo inicia-se com a concepção, fase que tem como principais objetivos a compreensão da oportunidade de criação de um programa de TVDI e a definição do que este programa se propõe a apresentar ao usuário/telespectador. As seguintes atividades são realizadas nesta fase: busca de oportunidades, desenvolvimento do projeto inicial e teste de concepção. Para que o programa concebido dê bons resultados, é necessário primeiramente buscar uma nova oportunidade, perseguindo algo diferente do que se tenha hoje na programação de TV convencional e utilizando todo o potencial disponível nos recursos de interatividade digital que possa surpreender o usuário/telespectador. A idéia central do programa, as intenções de serviços e as funcionalidades a serem oferecidas devem estar contidas no projeto inicial, que deverá ser validado ainda nesta fase.

A fase de elaboração consiste basicamente em estimar diversos fatores que podem afetar o desenvolvimento do programa de TVDI. As tarefas relacionadas a esta fase incluem o planejamento, onde é definida a equipe e são estimados os custos e o cronograma; a

roteirização, que deve conter, além de uma descrição textual do roteiro do programa, indicações que denotem as necessidades especiais da aplicação interativa; e a criação do projeto arquitetural e visual (disposição dos elementos gráficos) do software.

A fase seguinte é a de construção do programa de TVDI, onde todas as mídias utilizadas devem ser construídas e a aplicação interativa implementada. Desta forma, esta fase abrange a produção e codificação visual/gráfica (mídias), a codificação do programa de software, a produção do programa de TV e, por fim, a integração do mesmo à aplicação interativa, compondo o programa de TVDI.

A fase de teste é crucial para a garantia da qualidade do programa de TVDI. Se os testes forem conduzidos de maneira bem-sucedida, os erros no programa interativo serão descobertos antes de chegarem à casa do telespectador/usuário, o que evidencia a extrema importância desta fase para o sucesso do programa de TVDI. Os testes devem demonstrar que as funções da aplicação parecem estar funcionando de acordo com as especificações e que os requisitos de comportamento e desempenho parecem estar sendo satisfeitos. A Tabela 3, mostra alguns testes (técnicas, níveis e tipos) que, em conjunto, fornecem uma boa indicação da confiabilidade e qualidade do programa de TVDI a ser avaliado.

Tabela 3: Diversos testes utilizados na produção de programas de TVDI. (GAWLINSKI, 2003), com adaptações.

Teste	Descrição
Avaliação da interface gráfica	Testes realizados durante a atividade de programação visual para a escolha da interface e elementos gráficos que serão utilizados na aplicação. Os designers avaliam na TV os modelos produzidos e definem a interface final da aplicação.
Funcional	Verifica se a aplicação se comporta conforme especificado.
Unitário	Visam identificar possíveis falhas/erros no menor componente da aplicação de maneira isolada. Este componente pode vir a ser um método, uma classe ou qualquer outra unidade atômica de qualquer elemento do programa de TVDI.
Integração	Verifica se, integrados, os diferentes elementos (componentes ou serviços) do programa funcionam corretamente.

Teste	Descrição
Sistema	Verifica se o sistema como um todo (aplicação, <i>back-end</i> , plataformas utilizadas) funciona conforme esperado e “não faz nada que não deveria fazer”.
Aceitação	Valida se o produto final está de acordo com as expectativas do cliente.
Regressão	Re-teste após modificações nas aplicações que não passaram em outros testes.
Funcionalidade	Simula todos os cenários de uso do software e garante que todos os requisitos funcionais estejam implementados em conformidade.
Interface	Testa todas as interfaces existentes no sistema.
Desempenho	Verifica o desempenho da aplicação em diferentes circunstâncias e sob severas requisições ao serviço de interatividade. Contempla os testes de carga e de stress.
Usabilidade	Verifica a atratividade e facilidade de uso da aplicação/programa. Utiliza parâmetros como: tempo gasto para execução de tarefas pré-definidas, o caminho percorrido na tela, facilidade de aprendizado, taxa de erros, tempo de retenção do aprendizado, satisfação do usuário, etc.
Revisão tipográfica	Verifica as falhas gramaticais, a tipografia aplicada e outros possíveis agentes de erro dispostos nas mídias vinculadas ao programa de TVDI.

Ao finalizar a fase de teste com êxito, o programa está pronto para ser veiculado (fase de liberação). Uma vez o programa de TVDI disponível para veiculação, cabe ao produtor promover comercialmente o programa junto aos usuários/telespectadores fazendo a difusão do objetivo principal do programa, bem como analisar a audiência do programa, avaliar a reação do público em relação ao programa e a interatividade do mesmo. Estas atividades, para o produto do programa de TVDI são incessantes em toda a vida do programa.

Convém, por fim, ressaltarmos um diferencial no modelo de processo de produção de programas para TVDI: os *stakeholders* envolvidos. O sucesso das fases que permeiam este processo só pode ser garantido através da colaboração de uma equipe multidisciplinar, de forma que novos papéis foram adicionados às tradicionais equipes técnicas e de produção de programas para TV convencional.

Em seu trabalho, (VEIGA, 2006) descreve estas novas equipes, suas responsabilidades e os profissionais envolvidos. Um resumo pode ser observado na Tabela 4.

Tabela 4: Equipes envolvidas no processo de produção de programas de TVDI.

EQUIPE DE PRODUÇÃO	
Responsabilidades	Papéis
Responsável pela definição do programa, criação do <i>storyboard</i> , produção do projeto do programa interativo de forma preliminar e a entrega da aplicação final.	<ul style="list-style-type: none"> ▪ Produtor/gerente do projeto ▪ Usuário ▪ Assistente de TV ▪ Assistente de produção
EQUIPE DE PROJETO GRÁFICO	
Responsabilidades	Papéis
Responsável pela elaboração do projeto visual/gráfico da interatividade do programa.	<ul style="list-style-type: none"> ▪ Diretor de criação ▪ Projetista gráfico
EQUIPE TÉCNICA DE SOFTWARE	
Responsabilidades	Papéis
Responsável pela codificação do software de interatividade e unir o programa codificado com o programa convencional	<ul style="list-style-type: none"> ▪ Engenheiro de software ▪ Arquiteto de software ▪ Desenvolvedor ▪ Engenheiro de teste ▪ Integrador

2.5 DISCUSSÃO

Como vimos neste capítulo, conceitos não tão recentes como o da interatividade se uniram à modernidade da digitalização do sinal televisivo para criar uma nova mídia capaz de potencializar os serviços interativos na televisão, através da utilização da infraestrutura fornecida por um sistema de TVDI e seus componentes. Desta forma, uma gama de aplicações e programas interativos vem sendo desenvolvidos para os mais diversos fins e públicos, levando em consideração características como: relação do aplicativo com o conteúdo audiovisual do programa, tipo de interface utilizada e sua relação com o vídeo, diferentes possibilidades de interação com o usuário, linguagem de programação utilizada, entre diversas outras. Todas estas funcionalidades são fornecidas por meio de um *middleware* – no caso brasileiro, o Ginga - que provê todas as APIs necessárias à portabilidade destas aplicações nos diferentes equipamentos receptores.

A novidade por trás da execução de um software na televisão traz impactos não só na forma de assistir, mas também na forma de produzir este novo conteúdo. Assim, o processo de produção de programas para TV vem sendo modificado para dar suporte à produção de programas interativos, através da incorporação de atividades inerentes ao desenvolvimento de software, a exemplo da atividade de testes. Práticas como trabalho colaborativo e interativo, papéis multidisciplinares, modularidade e planejamento são essenciais para este novo modelo de processo.

Esta nova demanda por produção de conteúdo interativo e a exigência da integração de diversos atores multidisciplinares no processo tornam imprescindíveis a utilização de ferramentas de apoio ao desenvolvimento e a implantação de uma infraestrutura que permita a execução e testes de programas de TVDI, conforme discutido nos capítulos a seguir.

3

TRABALHOS RELACIONADOS

Uma das grandes dificuldades encontradas no desenvolvimento de conteúdo para TVDI diz respeito à necessidade de um ambiente que reproduza, da forma mais fiel possível, a execução e testes das aplicações e programas interativos, na tentativa de garantir a qualidade dos mesmos antes de sua exibição aos telespectadores. De forma a tentar suprir esta necessidade, existem no mercado algumas ferramentas simuladoras de aplicações interativas baseadas nos diversos padrões de *middleware* utilizados mundialmente, disponíveis gratuita ou comercialmente.

Neste capítulo, abordamos algumas das ferramentas mais utilizadas e que mais se assemelham ao ambiente proposto neste trabalho, seja por permitirem a execução e o teste de aplicações imperativas desenvolvidas na linguagem Java (*Xlets*), ou por serem baseadas na especificação do *middleware* Ginga. Por fim, é realizada uma análise destas ferramentas, onde são elencadas suas principais características e deficiências.

3.1 XLETVIEW

O XleTView¹⁹ é o emulador de TV Digital para PC mais popular entre os desenvolvedores de Xlets. Esta ferramenta possui o código aberto sob a licença GNU GPL e, além de uma implementação de referência da API JavaTV, traz consigo implementações de

¹⁹ <http://www.xletview.org/>

outras APIs especificadas no padrão DVB MHP, como HAVi e DAViC. Como foi desenvolvido totalmente em Java, este emulador é independente de plataforma. Para utilizá-lo, basta que seja feito o download através do site e que os arquivos sejam descompactados.

Assim como as TVs possuem uma lista de canais disponíveis ao telespectador, o XleTView também oferece esta funcionalidade, realizada de uma forma mais restrita. O usuário deve informar quais canais estão disponíveis, editando o arquivo “config/channels.xml”. A configuração de canais no XleTView consiste basicamente na escolha de um nome (ou número) e de um plano de fundo para o canal, que pode ser uma imagem estática ou um arquivo de vídeo em formato AVI. Outros parâmetros configuráveis nesta ferramenta estão relacionados a características básicas como: disposição da aplicação na tela, configuração da *safearea* (posição e dimensão) e configuração das características e teclas de evento do controle remoto.

A tela principal do XleTView pode ser visualizada na Figura 21. Para rodar uma aplicação, basta selecioná-la através do menu “Applications”. Contudo, o XleTView deve ser previamente informado sobre quais aplicações estão disponíveis e onde está localizado o código-fonte das mesmas.

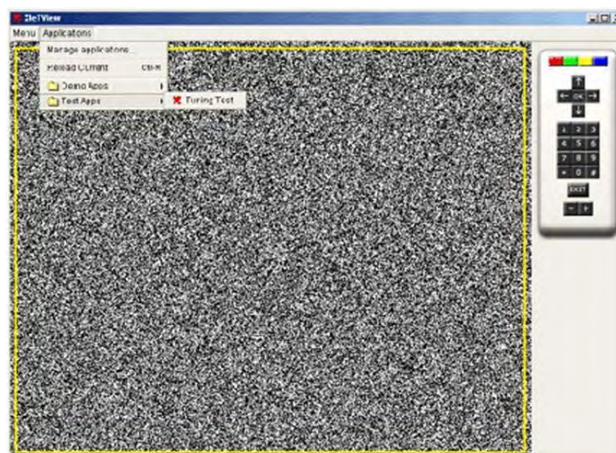


Figura 21: Tela principal do XleTView

Nos receptores MHP, as aplicações interativas são sinalizadas através da tabela AIT (*Application Information Table*), que contém todas as informações que o receptor de TVDI necessita para execução das mesmas. Isto não é possível com o XleTView, de forma que ele utiliza um Gerenciador de Aplicações (Figura 22) para definir aplicações ou grupos de

aplicações e suas características. Através do Gerenciador, algumas informações sobre a aplicação a ser testada devem ser preenchidas: o nome, o diretório raiz e a classe principal do *Xlet*. Trata-se de um subconjunto das propriedades que seriam sinalizadas através da AIT.

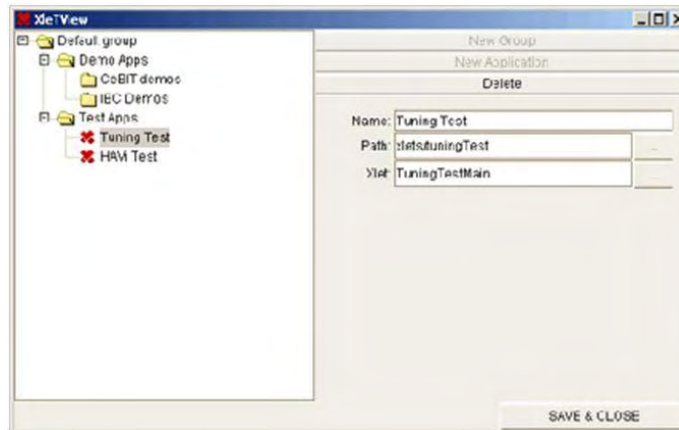


Figura 22: Gerenciador de Aplicações do XleTVIEW

Após devidamente configurada, a aplicação encontra-se acessível através do menu “Applications” do XleTVIEW. Ao ser selecionada, a aplicação é exibida na tela principal do emulador e o usuário pode interagir com a mesma através do controle remoto da ferramenta.

3.2 OPENMHP

O OpenMHP²⁰ é um projeto de software livre sob licença GNU LGPL 2.1 (*Lesser General Public Licence*) que simula, no computador, a execução de aplicações baseadas na especificação DVB-MHP. Por ser implementada em Java, é independente de plataforma, porém é requerida uma implementação da biblioteca JavaTV para o funcionamento da ferramenta.

A biblioteca OpenMHP é formada por dois componentes: o conjunto de classes MHP e a biblioteca da Camada de Adaptação (*Adaptation Layer*). A biblioteca MHP possui uma implementação independente de plataforma para a execução de aplicações MHP. A Camada de Adaptação é uma interface responsável pela interação da camada MHP com a camada específica de hardware do ambiente utilizado. Esta camada está apenas parcialmente implementada no OpenMHP.

²⁰ <http://www.openmhp.org>

Assim como no XletView, toda a configuração das aplicações a serem testadas deve ser realizada através do Gerenciador de Aplicações (Figura 23). Desta forma, torna-se necessário o preenchimento de algumas informações, tais como: nome do projeto, *classpath*, classe principal do Xlet, OrgID (*Organization ID*) e AppID (*Application ID*). Estes dois últimos campos estão relacionados à sinalização do contexto da aplicação e, em conjunto, formam um identificador único para a mesma. É possível também editar a forma de exibição das mensagens de erro e de *debug* informadas durante a realização dos testes. Todas estas configurações são realizadas através da aba de Propriedades do Gerenciador de Aplicações (Figura 24).

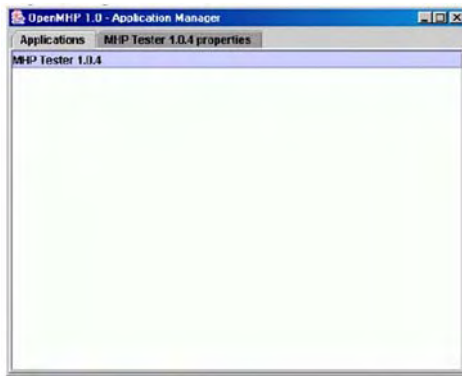


Figura 23: Gerenciador de Aplicações do OpenMHP

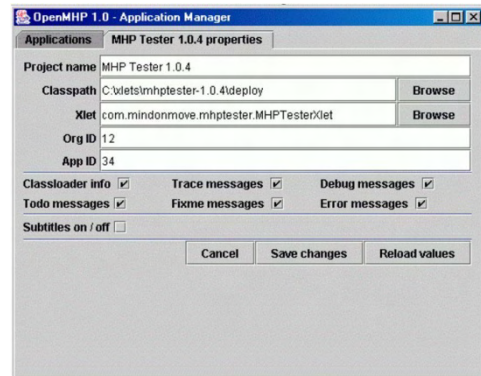


Figura 24: Tela de propriedades da aplicação no OpenMHP

Após devidamente configurada, a nova aplicação é exibida na lista de aplicações disponíveis. Ao ser selecionada, um novo processo de teste é iniciado e duas novas janelas são exibidas: uma para apresentação da saída gráfica da aplicação, formada pelos emuladores da tela da TV (*TVScreen*) e do controle-remoto, e outra para exibição da saída textual com as informações sobre o *debug* (Figura 25).

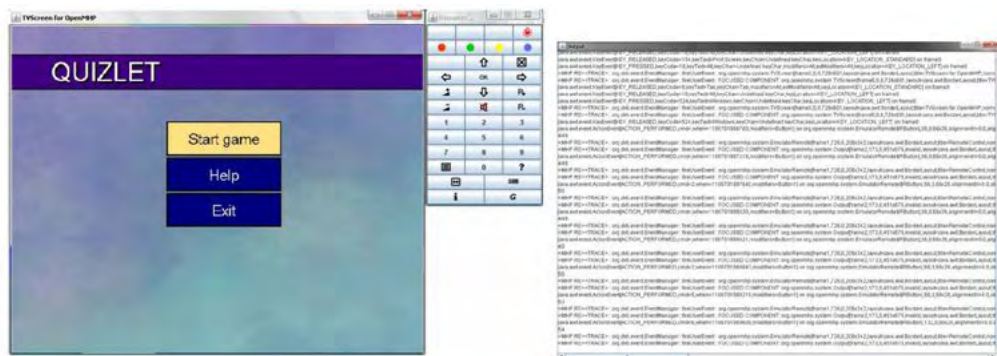


Figura 25: Janelas de saída gráfica e textual do OpenMHP

3.3 GINGA-NCL EMULATOR E SET-TOP BOX VIRTUAL GINGA-NCL

O Ginga-NCL Emulator²¹ é um emulador capaz de apresentar o conteúdo de uma aplicação de TV Digital escrita em linguagem NCL. Esta ferramenta desenvolvida na linguagem Java é dedicada a ambientes *desktop* e provê uma interface gráfica para a emulação de um ambiente de TV Digital baseado no *middleware* Ginga-NCL.

Para executar uma aplicação nesta ferramenta, basta escolher seu arquivo principal e então pressionar o botão para iniciá-la. Os componentes gráficos serão exibidos em uma nova janela e o usuário pode então interagir com a aplicação através de um emulador de controle remoto. A Figura 26 mostra uma aplicação sendo testada no Ginga-NCL Emulator.



Figura 26: Teste de aplicação NCL no Ginga-NCL Emulator

Já o STB Virtual Ginga-NCL²² é uma máquina virtual para VMWare²³ contendo a implementação de referência Ginga-NCL C++, executado sobre uma plataforma Linux. A versão C++ do Ginga-NCL foi concebida para ser uma implementação de maior desempenho, de forma que possa ser facilmente portada para plataformas com características semelhantes aos set-top boxes comerciais. A interface gráfica do STB Virtual é mostrada na Figura 27.

Todo o acionamento de comandos do set-top box virtual deve ser feito por meio de uma aplicação de console remoto SSH (*Secure Shell*). Somente as ações de interatividade serão feitas diretamente através da janela da máquina virtual.

²¹ <http://www.gingancl.org.br/ferramentas.html>

²² <http://www.gingancl.org.br/ferramentas.html>

²³ <http://www.vmware.com/>

O STB virtual é distribuído com duas aplicações NCL de exemplo. Para carregar outras aplicações é necessária a utilização de uma aplicação de transferência remota de arquivos. A Figura 28 exibe uma aplicação de exemplo rodando no STB Virtual.



Figura 27: Interface gráfica do STB Virtual Ginga-NCL



Figura 28: Teste de aplicação no STB Virtual Ginga-NCL

3.4 IRT MIDDLEWARE E IRT MHP TEST CENTRE

O IRT Middleware²⁴ é uma implementação comercial de um emulador para o *middleware* MHP, desenvolvido pela empresa alemã IRT - *Institut für Rundfunktechnik GmbH*, instituto de pesquisa privado responsável pelo desenvolvimento do padrão em conformidade com o sistema MHP utilizado pelos canais públicos de transmissão de TVDI alemão.

Este emulador possui todas as APIs da especificação MHP implementadas, de forma que as aplicações podem ser testadas em um ambiente mais fiel ao STB. Além disso, o software dá suporte a outras funcionalidades avançadas como: interpretação de um MPEG2 TS, que suporta multiplexação de instâncias de vídeo, áudio e dados; suporte a hardware de recepção de sinal de TVDI; sintonia de serviços; e emulação do carrossel de objetos. A Figura 29 ilustra a interface gráfica desta ferramenta.

²⁴ <http://www.irt.de/IRT/mhp/mhp-e.htm>

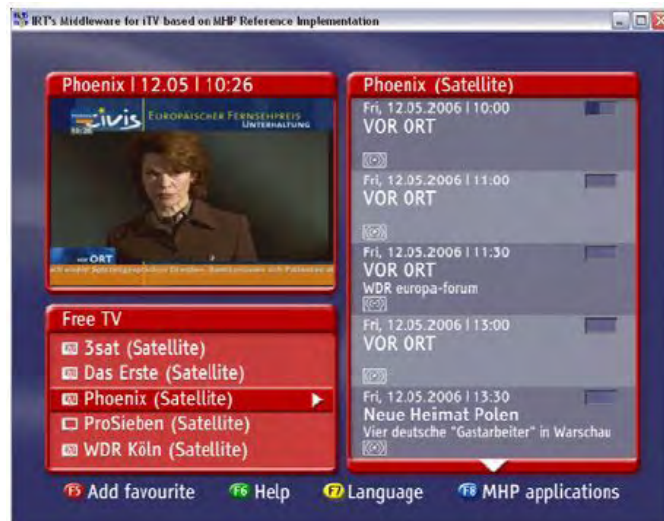


Figura 29: Interface do IRT Middleware

Com base no emulador de *middleware* IRT e em outras ferramentas próprias, a IRT disponibiliza, através do projeto MHP *Knowledge Database* (MHP-KDB)²⁵, um centro de testes de aplicações MHP, denominado IRT Test Centre.

O IRT Test Centre fornece, por meio de uma interface *web*, a infraestrutura necessária para a simulação de um ambiente de TVDI, desde a transmissão de conteúdo até sua recepção e exibição. O usuário realiza o *upload* de seus arquivos e visualiza o resultado dos seus testes em alguns dos receptores disponíveis, controlando-os remotamente e comparando o comportamento da aplicação nas diferentes plataformas (GERTZEN, 2005) (MHP-KDB, 2004).

A Figura 30 exibe a interface do IRT Test Centre que está dividida em frames: o de seleção do STB; o de apresentação da saída gráfica (vídeo e aplicação); o de exibição do controle remoto; o das informações sobre o STB; o do gerenciador de arquivos; o da saída textual (*debug*); e, finalmente, o que controla todo o processo de teste.

²⁵ <http://www.mhp-knowledgebase.org/>



Figura 30: Interface do ITR Test Centre

É necessário frisar que por trás deste centro de testes existe uma poderosa infraestrutura física, composta por diversos equipamentos caros, como mostra a Figura 31.



Figura 31: Parte da infraestrutura utilizada pelo IRT MHP Test Centre. Fonte: (IRT, 2005)

3.5 ANÁLISE COMPARATIVA DAS FERRAMENTAS

Os emuladores MHP gratuitos, apesar de constituírem uma boa opção para os iniciantes no desenvolvimento de aplicações para TVDI e possuírem uma interface simples e enxuta, possuem muitas restrições que merecem ser destacadas. Faz-se necessária também a observação de que, a despeito de sua popularidade, tanto o XleTView quanto o OpenMHP

tiveram seu desenvolvimento descontinuado e possuem implementações incompletas da especificação MHP e do conjunto de classes que o compõe.

Um dos maiores problemas do XleTView diz respeito a sua restrita possibilidade de utilização de vídeos. Esta ferramenta utiliza a biblioteca JMF 1.0 (*Java Media Framework*) da SUN que, por ser totalmente implementada em Java, não possui um bom desempenho na execução de vídeos. Além disso, o XleTView não suporta vídeos no formato MPEG, permitindo apenas a reprodução de arquivos AVI codificados com o codec Cinepak. Também não são possíveis: o carregamento de aplicações a partir de um *transport stream*; o suporte a carrossel de dados; a comunicação entre *xlets*; o acesso a informações de serviços; o controle de acesso; entre outras funcionalidades necessárias à execução de aplicações mais avançadas e testes reais (MORRIS, 2008).

Já o emulador OpenMHP não suporta a reprodução de nenhum tipo de vídeo. Os desenvolvedores desta ferramenta recomendam a utilização uma imagem estática como plano de fundo da aplicação e o uso de técnicas de redimensionamento como forma de simulação de um vídeo (ArviD, 2004). As deficiências encontradas nesta ferramenta são basicamente as mesmas listadas para o XleTView. Os problemas na execução de vídeos e a impossibilidade de tratamento de eventos relacionados ao conteúdo audiovisual e às aplicações impede o teste de programas interativos nestes emuladores.

Com relação aos ambientes de teste para o *middleware* Ginga, temos atualmente a nossa disposição apenas emuladores relacionados à implementação declarativa Ginga-NCL. O Ginga-NCL Emulator é uma ferramenta fácil de usar e que deve ser utilizada para testes de programas NCL simples, funcionando bem para fins didáticos. Ela não oferece suporte a *alfa blending*, transparência, efeitos de transição, entre outros, além de possuir limitações na reprodução de vídeos.

Já o STB Virtual Ginga-NCL possui recursos mais avançados de apresentação de aplicações declarativas, melhor desempenho em relação ao emulador implementado em Java e maior proximidade de uma implementação embarcada em set-top boxes reais. Para sua instalação e utilização, é necessária boa experiência em ambiente Linux. Os reportes dos

membros da comunidade Ginga²⁶ mostram a dificuldade enfrentada por usuários não-avançados para colocar essa versão do Ginga-NCL em funcionamento em seus computadores. Mas, ao mesmo tempo, vê-se que a demanda pelo uso da versão C++ vem crescendo, provavelmente motivada pela maior fidelidade de apresentação dos documentos NCL, se comparada à versão escrita em Java. Uma restrição desta ferramenta diz respeito à limitação da máquina virtual a uma resolução 640X480, o que impede testes de aplicativos para alta definição.

O IRT Middleware é uma das ferramentas mais utilizadas no mercado comercial, devido à sua implementação completa do MHP e ao seu bom desempenho na execução das aplicações. Um ponto negativo é o alto valor da licença desta ferramenta, o que impossibilita que a grande maioria dos desenvolvedores tenha acesso à mesma.

O IRT Test Centre também fornece uma boa solução de testes de aplicações MHP, principalmente por possuir ferramentas de transmissão do conteúdo, e não só de exibição do mesmo. Todas as pessoas registradas no site MHP-KDB podem ter acesso a interface *web* e se comprometem a publicar todas as informações adquiridas através do uso desta ferramenta. Ao contrário das outras ferramentas analisadas, esta não utiliza uma infraestrutura própria, e sim de terceiros, de forma que todos os testes devem ser agendados com no mínimo dois meses de antecedência. Vale salientar o alto custo de implantação e manutenção de uma infraestrutura como esta.

²⁶ http://www.softwarepublico.gov.br/dotlrn/register?community_id=1101545&referer=/dotlrn/clubs/ginga/

4

O AMBIENTE XTATION

Neste capítulo apresentamos o ambiente alvo deste trabalho, através da descrição do sistema como um todo, da sua arquitetura e das ferramentas que o compõem; do detalhamento de sua aplicabilidade no processo de produção de conteúdo para TVDI e no processo de testes de componentes para o *middleware* utilizado; bem como da descrição de todo o processo de desenvolvimento e de suas fases de análise, projeto e implementação.

4.1 VISÃO GERAL

O Xtation é um ambiente para execução e testes de aplicações baseadas no *middleware* Ginga, que visa facilitar o processo de testes em um ambiente de baixo custo e com características bem próximas a de um sistema de TVDI real. A sua estrutura deve proporcionar a geração e a transmissão de fluxos contínuos de vídeo, áudio e dados multiplexados, tal qual seria em uma estação de TVDI, bem como sua recepção e correta exibição no televisor acoplado ao ambiente.

Este ambiente possui uma arquitetura distribuída em duas estações, denominadas “Servidor de Vídeo e Aplicação” e “Servidor de Middleware”, conforme pode ser observado na Figura 32. Estas estações não necessitam de monitores, pois deverão ser controladas e configuradas por meio de uma aplicação remota, denominada XtationWizard, executada a

partir de uma estação cliente. Esta aplicação deve ser simples e intuitiva, a fim de permitir que qualquer usuário a utilize e teste seus programas de TVDI. O sistema deve manter o usuário informado sobre o que está acontecendo durante o processo de teste, bem como sobre o *log* gerado na execução das aplicações.

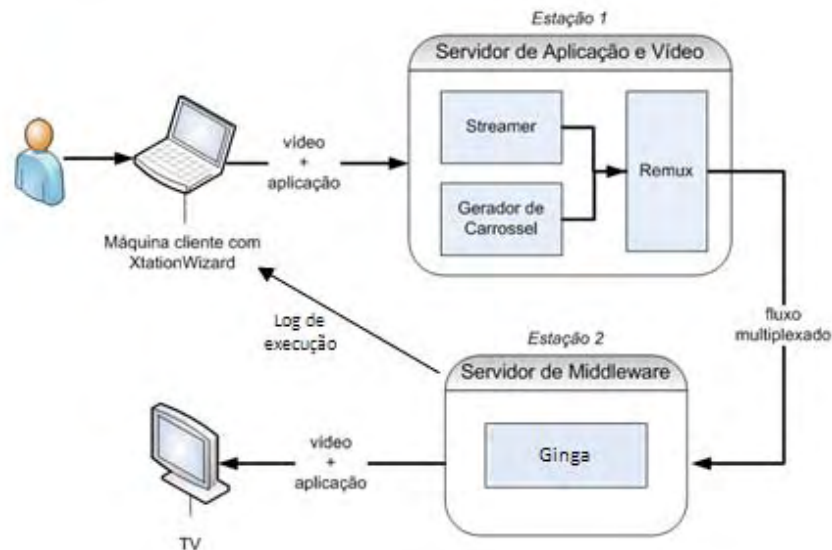


Figura 32: Arquitetura do ambiente Xtation

4.2 COMPONENTES

Como pode ser observado na Figura 32, a arquitetura do Xtation é baseada em componentes específicos para geração, transmissão e recepção dos fluxos dos programas interativos. Nas subseções seguintes são detalhados cada um desses componentes.

4.2.1 Streamer

O Streamer é uma aplicação responsável por enviar um fluxo em formato MPEG TS para um ou mais computadores em uma rede, com o compromisso de manter na saída a mesma taxa em que o fluxo deve ser consumido na outra ponta (receptor). Como vantagens, esta técnica evita que o receptor tenha que armazenar localmente o fluxo recebido, além de promover a simulação de uma estação de TV, que sempre está transmitindo na taxa correta em que o vídeo deve ser consumido. O Streamer deve receber como parâmetros o arquivo a

ser transmitido, bem como o endereço e a porta do receptor (endereçamento IP). A saída do Streamer é uma das entradas do Remux (item 4.2.3).

4.2.2 Gerador de Carrossel

O objetivo do Gerador de Carrossel é transformar um conjunto de arquivos de dados em um fluxo elementar, empregando um esquema de transmissão cíclica. Para tanto, todo o sistema de arquivos é armazenado em um arquivo de Carrossel de Objetos (OC). Ao sintonizar um determinado canal, o receptor deve ser capaz de decodificar os dados recebidos e colocá-los em uma área de memória para que possam ser utilizados, preservando a estrutura de arquivos e diretórios enviada. No ambiente Xtation é utilizado o Gerador de Carrossel LinuxTV²⁷. A saída desta ferramenta (o Carrossel de Dados) é uma das entradas do Remux.

4.2.3 Remux - Multiplexador

O Remux é uma aplicação responsável por multiplexar em um mesmo fluxo MPEG TS, vários fluxos elementares (como fluxos de vídeo, áudio e dados), através da intercalação dos pacotes destes fluxos. Esta ferramenta possui N entradas e uma saída, podendo cada fluxo ser inserido em uma entrada diferente, ou podendo também ser inserido um fluxo misto por uma única entrada. Não há limitações sobre a quantidade de fluxos a serem inseridos, podendo haver vários fluxos de vídeo ou aplicação ao mesmo tempo, devendo, contudo, a banda de transmissão ser levada em consideração. A taxa de transmissão dos fluxos de áudio e vídeo é mantida constante, já a taxa de transmissão dos fluxos de aplicação e dados é mais flexível, sendo regulada de acordo com o limite da banda de transmissão.

A saída do Remux (fluxo multiplexado) é enviada via rede IP ao Servidor de Middleware, onde é devidamente manipulada.

4.2.4 Servidor de Middleware

O Servidor de Middleware tem como principal componente a implementação do *middleware* Ginga utilizada para a execução dos programas de TVDI, neste caso o OpenGinga. Por ser totalmente baseado na especificação Ginga, esta implementação possui

²⁷ LinuxTV - Television with Linux. <http://www.linuxtv.org/>

em sua arquitetura uma camada composta por decodificadores e procedimentos comuns para obter conteúdo transportado em fluxos de transporte MPEG TS ou através de um canal de retorno. Além disso, o OpenGinga oferece todo o suporte para a execução de aplicações procedurais e para a exibição de vários objetos de mídia. Maiores detalhes sobre sua arquitetura são descritos no item 2.2.

Vale salientar que, para este ambiente, o *middleware* é visto como uma “caixa-preta”. Por possuir uma arquitetura flexível e adaptativa, não há limitações quanto ao uso de outra implementação do *middleware* Ginga como parte integrante da arquitetura utilizada.

4.2.5 XtationWizard

O XtationWizard é uma ferramenta desenvolvida em Java, com o intuito de controlar e guiar todo o processo de execução e teste dos programas interativos. Esta ferramenta deve ser instalada na máquina cliente, atuando como a única interface entre o usuário e o ambiente Xtation. Para o usuário final, todas as demais ferramentas instaladas no Servidor de Aplicação e Vídeo e no Servidor de Middleware são transparentes, sendo apenas seus endereços de rede informados no XtationWizard.

É através do XtationWizard que o usuário insere todas as informações sobre o vídeo a aplicação a serem testados, configura e acompanha todo o processo de execução e teste, e visualiza o *log* de execução de seu programa de TVDI.

4.3 APLICABILIDADE

A aplicabilidade do Xtation é analisada mediante o contexto do processo de desenvolvimento de aplicações e programas para TVDI. Vimos no item 2.4 a importância da etapa de testes na produção de conteúdo interativo, bem como os diversos tipos de testes utilizados. A Figura 33 mostra as atividades onde o Xtation pode estar inserido dentro do processo de produção analisado. Já a Figura 34 expande estas atividades e nos mostra algumas possibilidades de utilização do ambiente proposto como forma de auxiliar a execução de testes durante as diversas fases do processo de produção de programas de TVDI.

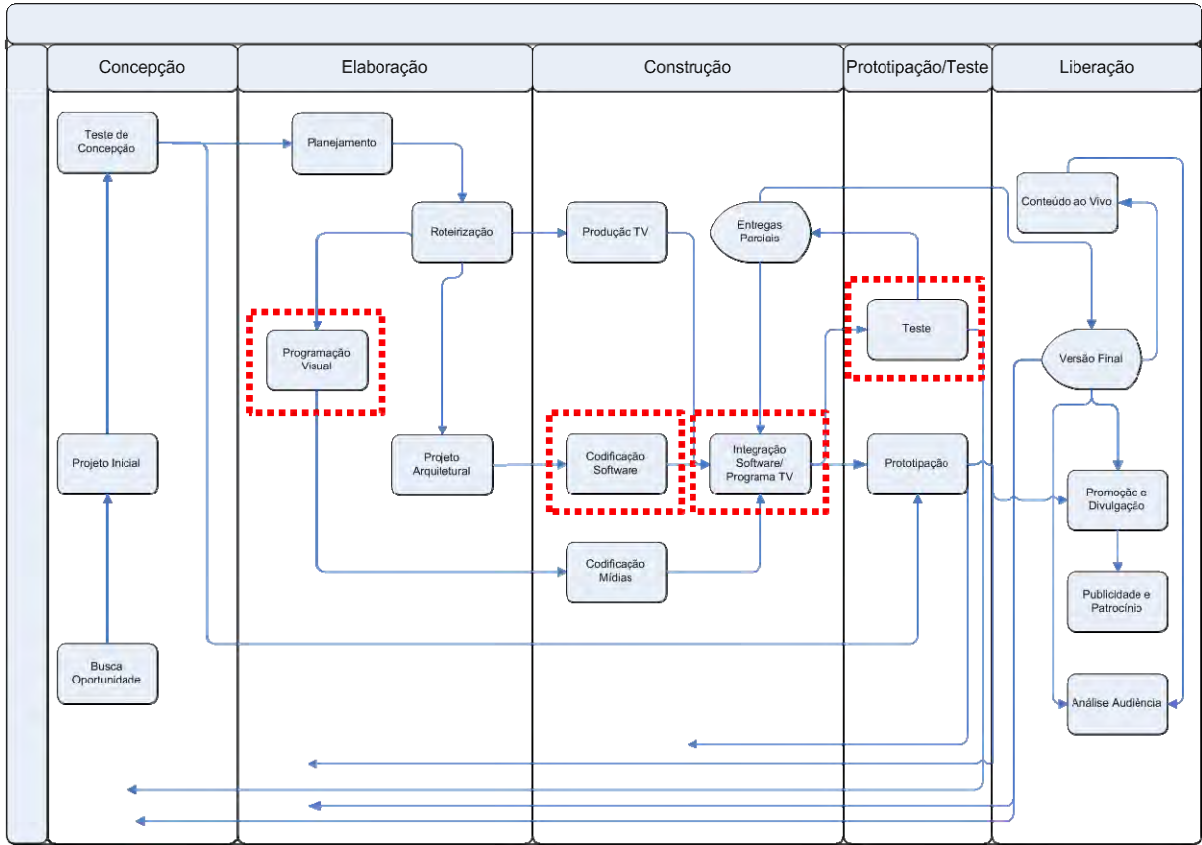


Figura 33: Atividades do processo de produção de programas de TVDI onde o Xtation está inserido.

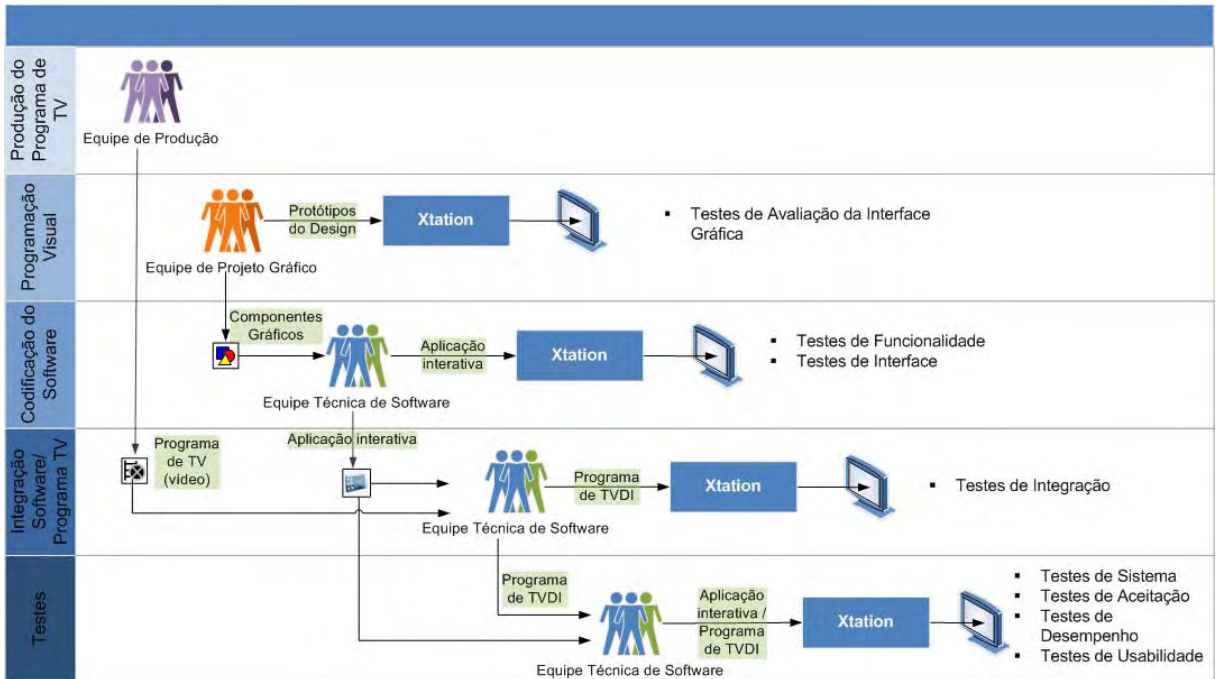


Figura 34: Aplicabilidade do Xtation no processo de produção de programas de TVDI

Durante a fase de Programação Visual, a equipe de projeto gráfico é responsável por desenvolver alguns protótipos para o design das aplicações interativas. É necessário testar os componentes gráficos diretamente em um ambiente de TV, tanto pela diferença significativa entre a exibição de gráficos no monitor e na TV, quanto pelas diferenças ocasionadas pela utilização das APIs de renderização gráfica do *middleware*. Nesta etapa, podemos utilizar uma simples aplicação para exibir as imagens dos protótipos desenvolvidos. Através do Xtation, esta aplicação é enviada ao *middleware* e exibida na TV, facilitando a realização dos Testes de Avaliação da Interface Gráfica.

Após a definição da interface gráfica final, os componentes gráficos (como imagem de fundo, ícones de navegação, etc) devem ser convertidos para um formato suportado pelo *middleware*, como por exemplo, JPEG ou GIF. Os arquivos finais serão então enviados à equipe técnica de software que deverá incorporá-los à aplicação durante a fase de Codificação, onde ocorre tanto a implementação da interface gráfica quanto dos requisitos funcionais do software. Através do Xtation podemos realizar os Testes de Funcionalidade e de Interface, para garantir a conformidade da aplicação em relação a sua especificação.

Programa de TV finalizado pela equipe de produção e aplicação interativa implementada e testada, seguiremos para a fase de Integração do software com o programa de TV. Novamente, a equipe técnica utilizará o Xtation na realização dos Testes de Integração.

Após a integração bem sucedida, teremos finalmente o programa de TVDI que precisa passar por uma série de testes que comprovem sua qualidade, dentre eles: o Teste de Sistema, o Teste de Aceitação, o Teste de Desempenho e o Teste de Usabilidade. O Xtation pode também ser utilizado como ferramenta facilitadora durante este processo.

4.4 ANÁLISE E PROJETO DO XTATION

Na realização das etapas de análise e projeto da ferramenta proposta foi utilizada a metodologia proposta por (LARMAN, 2002) bem como a linguagem de modelagem UML (*Unified Modeling Language*) (JACOBSON, et al., 2005) (GUEDES, 2006) que descreve

modelos de sistema - do mundo real e de software – baseado em conceitos de objetos. Desta forma, apresentamos a especificação de requisitos, o projeto da arquitetura, detalhes de implementação, a visão de implantação e, por fim, o projeto de interface com o usuário.

4.4.1 Especificação de Requisitos e Visão de Casos de Uso

De forma a facilitar a compreensão do comportamento externo do sistema e uma visão geral das funções e serviços fornecidos aos usuários, a Figura 35 apresenta o diagrama de casos de uso do Xtation.

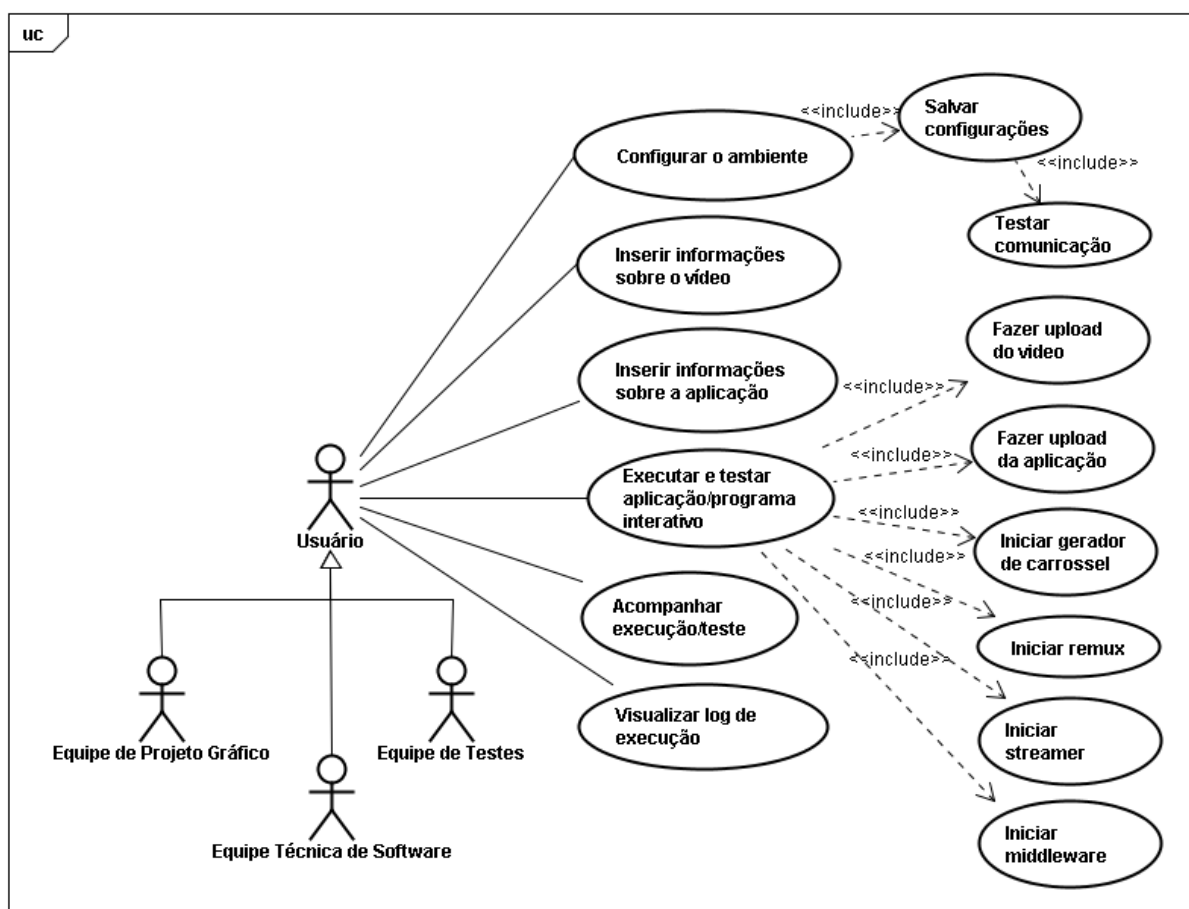


Figura 35: Diagrama de Casos de Uso UML do Xtation.

Os requisitos funcionais do ambiente, classificados em evidentes e ocultos, estão listados na Tabela 5. Enquanto os requisitos funcionais evidentes são efetuados com conhecimento do usuário, os requisitos funcionais ocultos são efetuados pelo sistema sem o conhecimento explícito do mesmo. Por sua vez, os requisitos não-funcionais (restrições

colocadas sobre como o sistema deve realizar seus requisitos funcionais) estão listados na Tabela 6.

Tabela 5: Requisitos funcionais do Xtation

Requisitos Funcionais		
Ref. #	Nome	Categoria
F1	Permitir que o usuário configure o ambiente	Evidente
F2	Salvar configurações do ambiente	Evidente
F3	Testar comunicação entre os servidores	Evidente
F4	Coletar informações sobre o vídeo	Evidente
F5	Coletar informações sobre a aplicação	Evidente
F6	Executar e testar aplicação/programa interativo	Evidente
F6.1	Fazer upload do vídeo	Oculto
F6.2	Fazer upload da aplicação	Oculto
F6.3	Iniciar Gerador de Carrossel	Oculto
F6.4	Iniciar Remux	Oculto
F6.5	Iniciar Streamer	Oculto
F6.6	Iniciar Middleware	Oculto
F7	Exibir mensagens dos eventos realizados	Evidente
F8	Exibir log de execução	Evidente
F9	Exibir simulação da comunicação do controlador com as outras máquinas do ambiente	Evidente
F10	Controlar upload do vídeo e da aplicação	Oculto
F11	Fornecer documentação de ajuda ao usuário	Evidente
F12	Exibir a aplicação e o vídeo na televisão	Evidente
F13	Permitir o reinício do teste, utilizando as configurações e informações anteriores	Evidente

Tabela 6: Requisitos não-funcionais do Xtation

Requisitos Não-Funcionais		
Ref. #	Restrição	Categoria
NF1	A interface do sistema deve ser implementada como formulários simples e intuitivos	Interface
NF2	Uma simulação da comunicação entre as máquinas do ambiente e a execução de seus processos deve ser exibida, de modo a permitir o acompanhamento do teste pelo usuário	Interface

NF3	Em todas as telas, um sistema de ajuda ao usuário deve estar disponível	Usabilidade
NF4	O sistema deve gerenciar falha de comunicação entre as máquinas que compõem o ambiente	Confiabilidade
NF5	Os endereços das máquinas que compõem o ambiente devem ser configuráveis pelo usuário.	Configurabilidade
NF6	Uma camada de persistência deve ser implementada a fim de armazenar os arquivos de configuração do sistema.	Persistência
NF7	Por cada vez, só pode ser testada uma aplicação e um vídeo	Especificação
NF8	Poderão ser testadas aplicações em Java (Xlets)	Especificação
NF9	O sistema deve prover um teste para garantir que todas as máquinas estão configuradas corretamente e estão comunicáveis.	Confiabilidade
NF10	O sistema deve ser entregue empacotado ao usuário final e com simples instalação	Empacotamento

4.4.2 Expansão dos Casos de Uso

Visando uma compreensão mais profunda dos processos e requisitos do sistema, os principais casos de usos foram expandidos, conforme pode ser observado adiante.

Tabela 7: Caso de Uso Expandido: Configurar o ambiente

Caso de Uso:	Configurar o Ambiente
Atores:	Usuário
Requisitos Correlacionados:	F1, F2, F3, NF5, NF6
Finalidade:	Permitir que o usuário configure os endereços das máquinas que compõem o ambiente
Sequencia Típica de Eventos:	<ol style="list-style-type: none"> 1. O usuário informa o IP do Servidor de Vídeo e Aplicação 2. O usuário informa o IP do Servidor de Middleware 3. O sistema valida os endereços fornecidos pelo usuário 4. O sistema testa comunicação com as máquinas. 5. O sistema salva as configurações.
Tratamento de Exceções:	<ol style="list-style-type: none"> 3a. Erro na formatação do endereço <ol style="list-style-type: none"> 3a.1. O sistema exibe mensagem de erro 3a.2. O usuário informa novamente os endereços dos servidores 3a.3. Retorna ao fluxo principal no passo 3 4a. Erro na comunicação com as máquinas informadas <ol style="list-style-type: none"> 4a.1. O sistema exibe mensagem de erro 4a.2. O usuário informa novamente os endereços dos servidores 4a.3. Retorna ao fluxo principal no passo 4

Tabela 8: Caso de Uso Expandido: Inserir informações sobre o vídeo

Caso de Uso:	Inserir informações sobre o vídeo
Atores:	Usuário
Requisitos Correlacionados:	F4
Finalidade:	Coletar todas as informações relativas ao vídeo que será usado no teste.
Sequencia Típica de Eventos:	<ol style="list-style-type: none"> 1. O usuário informa o caminho do vídeo, por meio de janelas de seleção de arquivos. 2. O sistema valida informação do usuário

Tabela 9: Caso de Uso Expandido: Inserir informações sobre a aplicação

Caso de Uso:	Inserir informações sobre a aplicação
Atores:	Usuário
Requisitos Correlacionados:	F5
Finalidade:	Coletar todas as informações relativas à aplicação que será usada no teste.
Sequencia Típica de Eventos:	<ol style="list-style-type: none"> 1. O usuário informa o diretório onde se encontra a aplicação 2. O usuário informa o arquivo principal da aplicação 3. O usuário seleciona se a aplicação será auto-start ou não. 4. O sistema valida as informações do usuário

Tabela 10: Caso de Uso Expandido: Testar a aplicação

Caso de Uso:	Executar e testar aplicação/programa interativo
Atores:	Usuário
Requisitos Correlacionados:	F6, F6.1, F6.2, F6.3, F6.4, F6.5, F6.6, F7, F8, F9, F10, F12
Finalidade:	Permitir que o usuário visualize sua aplicação na televisão, rodando no middleware OpenGinga.
Sequencia Típica de Eventos:	<ol style="list-style-type: none"> 1. O usuário seleciona a opção de iniciar teste da aplicação 2. O sistema faz upload do vídeo para o Servidor de Vídeo e Aplicação 3. O sistema faz upload da aplicação para o Servidor de Vídeo e Aplicação 4. O sistema inicia remotamente o Gerador de Carrossel 5. O sistema inicia remotamente o Remux 6. O sistema inicia remotamente o Streamer 7. O sistema inicia remotamente o Middleware 8. O sistema exibe aplicação e vídeo na televisão

	[Durante todos os passos de 2 a 7, o sistema exibe para o usuário informações sobre o log do processo de teste]
Tratamento de Exceções:	2-7a. Erro durante o processo 2-7a.1. O sistema exibe mensagem de erro e pára o processo de teste 2-7a.2. Retorna ao fluxo principal no passo 1

4.4.3 Projeto da Arquitetura – Visão Lógica

O projeto arquitetural do Xtation é apresentado na Figura 36, onde um Diagrama de Pacotes ilustra como os principais módulos do ambiente estão divididos em agrupamentos lógicos (visão lógica) mostrando as dependências entre estes.

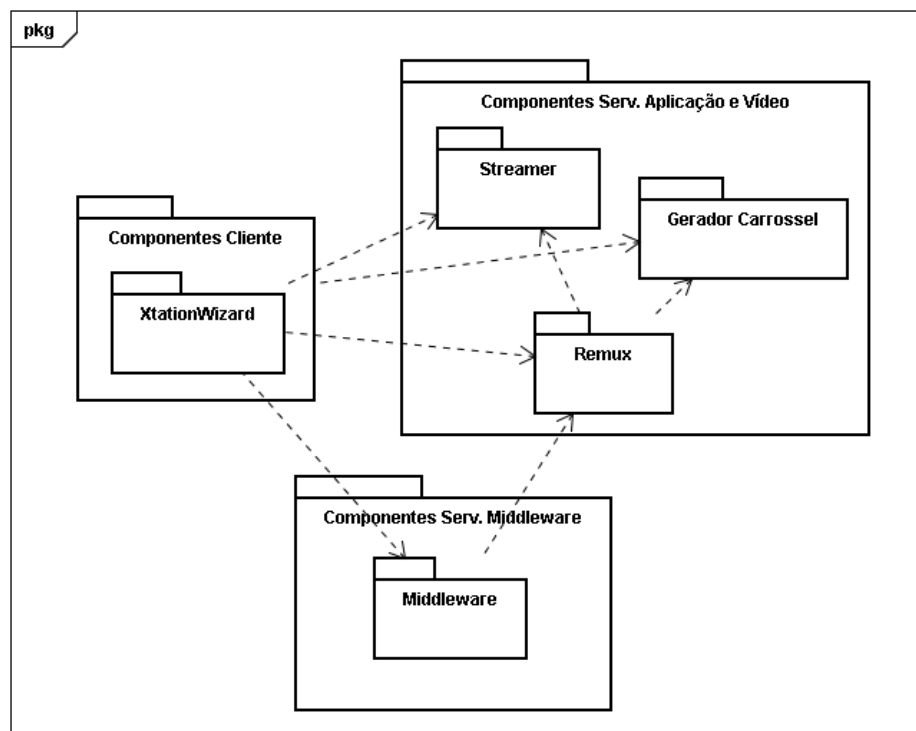


Figura 36: Diagrama de Pacotes do Xtation.

4.4.4 Visão de Implementação

O Diagrama de Classes exibido na Figura 37 revela detalhes da implementação do Xtation, definindo a estrutura das classes utilizadas pelo sistema, os atributos e métodos possuídos por cada classe, bem como o relacionamento entre elas.

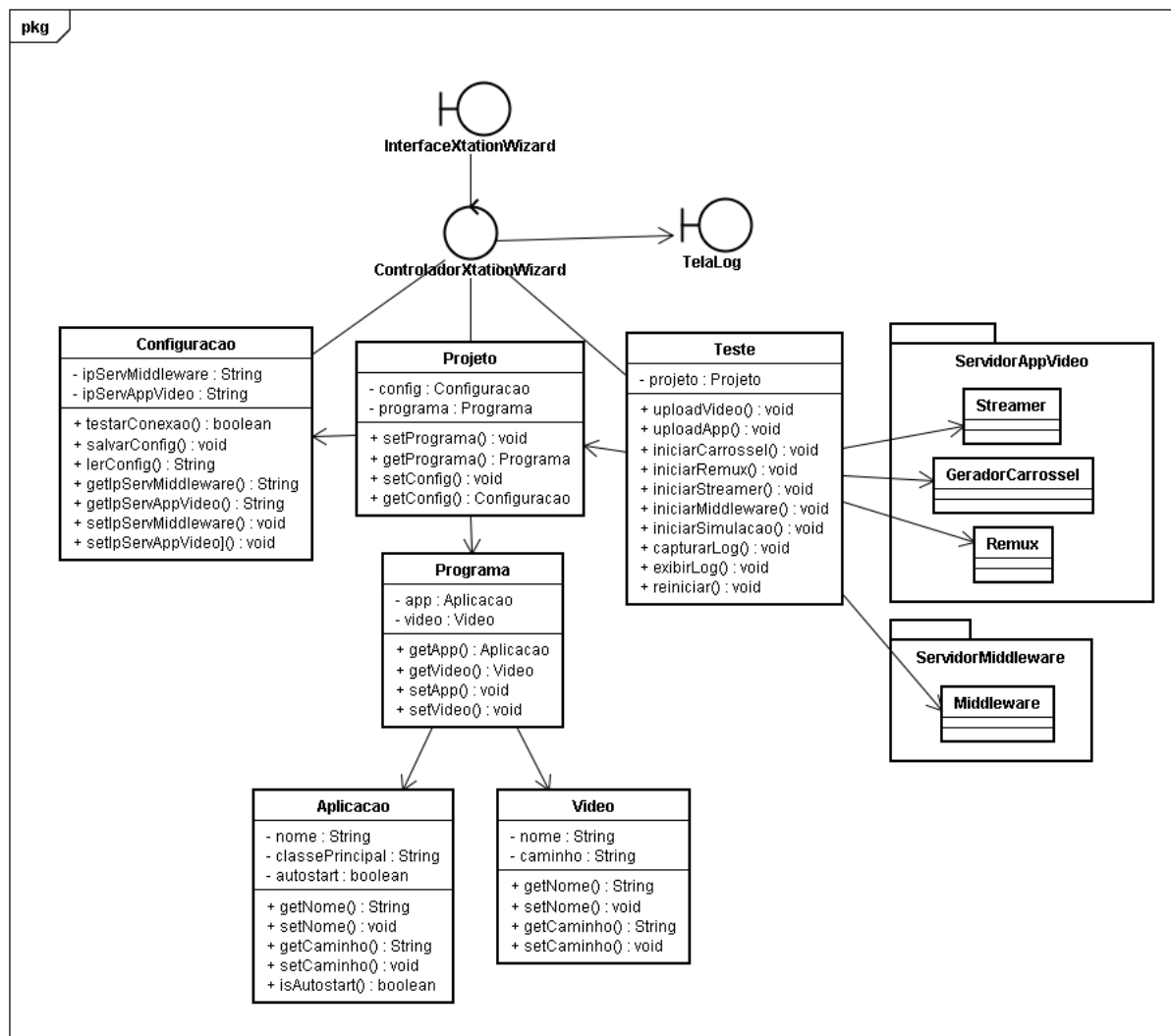


Figura 37: Diagrama de Classes do Xtation.

4.4.5 Visão de Implantação

O Diagrama de Implantação apresentado na Figura 38 enfoca a arquitetura distribuída e o ambiente físico onde é executado o Xtation, além de definir como as máquinas que compõem o ambiente estão conectadas e através de quais protocolos se comunicam.

A ferramenta XtationWizard, responsável por controlar o ambiente Xtation, comunica-se com as ferramentas do Servidor de Aplicação e Vídeo e do Servidor de Middleware através da interface de programação Java RMI (*Remote Method Invocation*) (DEITEL, et al., 2003), que permite a execução de chamadas remotas no estilo RPC (*Remote*

Procedure Call) em aplicações desenvolvidas em Java. Através da utilização da arquitetura RMI, é possível que um objeto ativo em uma máquina virtual Java possa interagir com objetos de outras máquinas virtuais Java, independentemente da localização das mesmas.

A comunicação entre os Servidores é realizada através do protocolo TCP/IP, enquanto o Servidor de Middleware conecta-se à televisão através de conectores DVI, RCA ou qualquer outra interface de vídeo, conforme disponibilidade nos equipamentos.

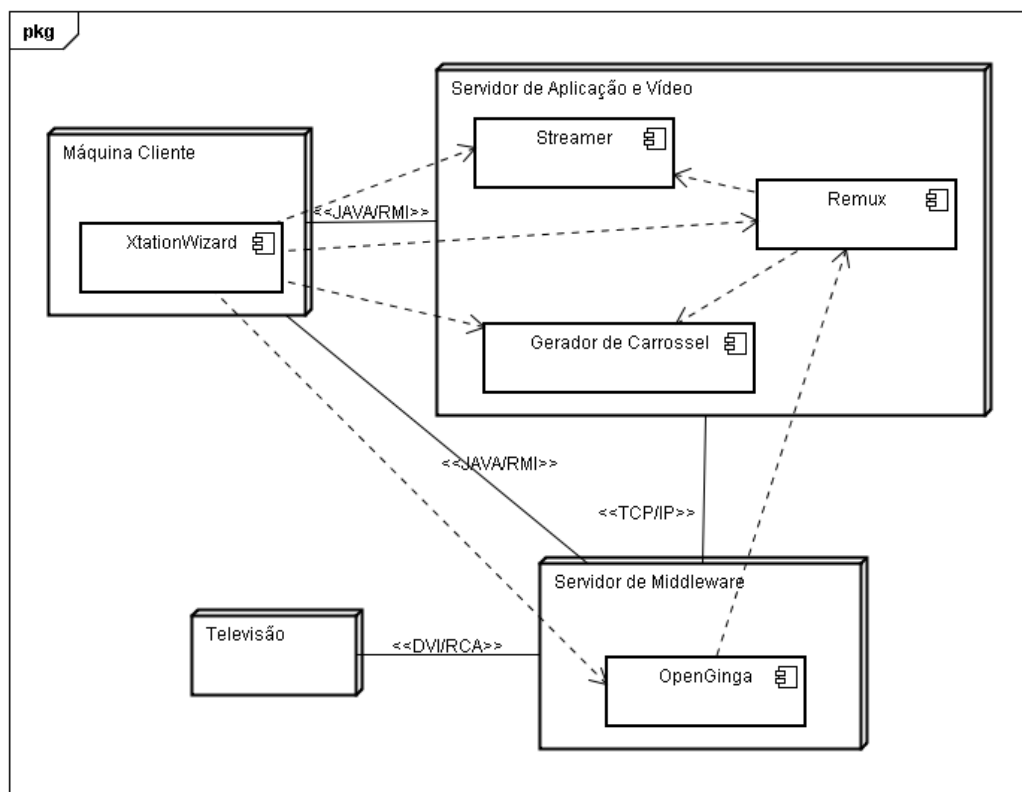


Figura 38: Diagrama de Implantação do Xtation.

4.4.6 Projeto da Interface com o Usuário

A interface da ferramenta controladora XtationWizard é ilustrada nesta subseção. Através da tela inicial (Figura 39), o usuário tem a opção de criar um “Novo Projeto” ou “Configurar” o ambiente, onde devem ser informados os endereços de rede do “Servidor de Aplicação e Vídeo” e do “Servidor de Middleware” (Figura 40). A ferramenta realizou um teste para verificar a comunicação entre a máquina cliente e os servidores configurados.



Figura 39: Tela do inicial do XtationWizard com as opções "Novo Projeto" e "Configurar".

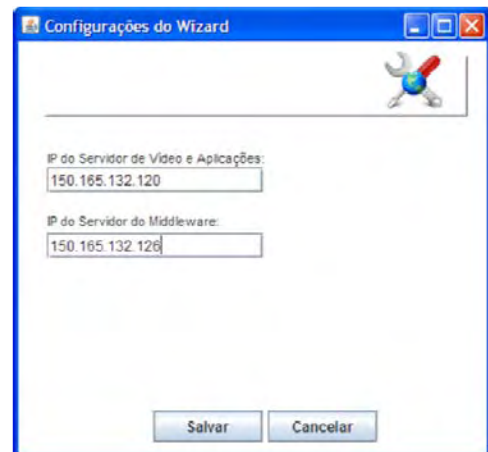


Figura 40: Tela de configuração do ambiente Xtation.

Ao clicar na opção “Novo Projeto”, é exibida a tela ilustrada na Figura 41, onde o usuário insere as informações sobre a aplicação e o vídeo a serem testados, preenchendo alguns campos como: a localização do arquivo de vídeo, o diretório base da aplicação, o caminho completo da classe principal do *Xlet* e se a aplicação é *autostart* ou não.



Figura 41: Tela do XtationWizard que coleta as informações sobre a aplicação e o vídeo

Após o preenchimento correto de todos os campos, são iniciadas as etapas que permitem a efetiva execução do programa interativo: *upload* da aplicação, *upload* do vídeo, início do carrossel de dados, início do *remux*, início do *streamer* e início do *middleware*. Todo

5

ESTUDO DE CASO

O estudo de caso descrito neste capítulo tem como finalidade a validação da proposta deste trabalho e foi desenvolvido de acordo com os possíveis cenários de utilização do ambiente Xtation, descritos no item “4.3 – Aplicabilidade”, e de acordo com as principais características das aplicações e programas de TVDI apresentadas no item 2.3.

5.1 A INFRAESTRUTURA

Para fins de validação do ambiente proposto, foi utilizada a infraestrutura física disponível no LAViD, onde foram utilizados equipamentos básicos como computadores, notebook e televisão, para a instalação e configuração de todas as ferramentas que compõem o Xtation.

Conforme descrito neste trabalho, o Xtation tem como principal atrativo seu baixo custo ao permitir a simulação de um ambiente de TVDI real, e para tanto utiliza-se de uma arquitetura distribuída composta por um conjunto de ferramentas instaladas em equipamentos simples e configurados em rede, como mostra a Figura 44.

Nesta infraestrutura, utilizamos um notebook para fazer o papel da máquina cliente, tendo como único requisito possuir máquina virtual Java instalada, a fim de permitir a correta execução do XtationWizard. As ferramentas que compõem o Servidor de Aplicação e Vídeo foram instaladas em um *desktop* básico executando plataforma *Linux*. Estas

ferramentas podem ser também instaladas em ambiente *Windows* e todas as bibliotecas necessárias para sua execução são configuradas no momento da instalação. Para o Servidor de Middleware, foi utilizado um computador compacto (MiniPC), onde encontra-se instalada a versão “v01_alpha” do *middleware* Openginga, executada em plataforma *Linux* (Ubuntu 9.10). As especificações básicas de todos estes equipamentos estão detalhadas na Tabela 11.

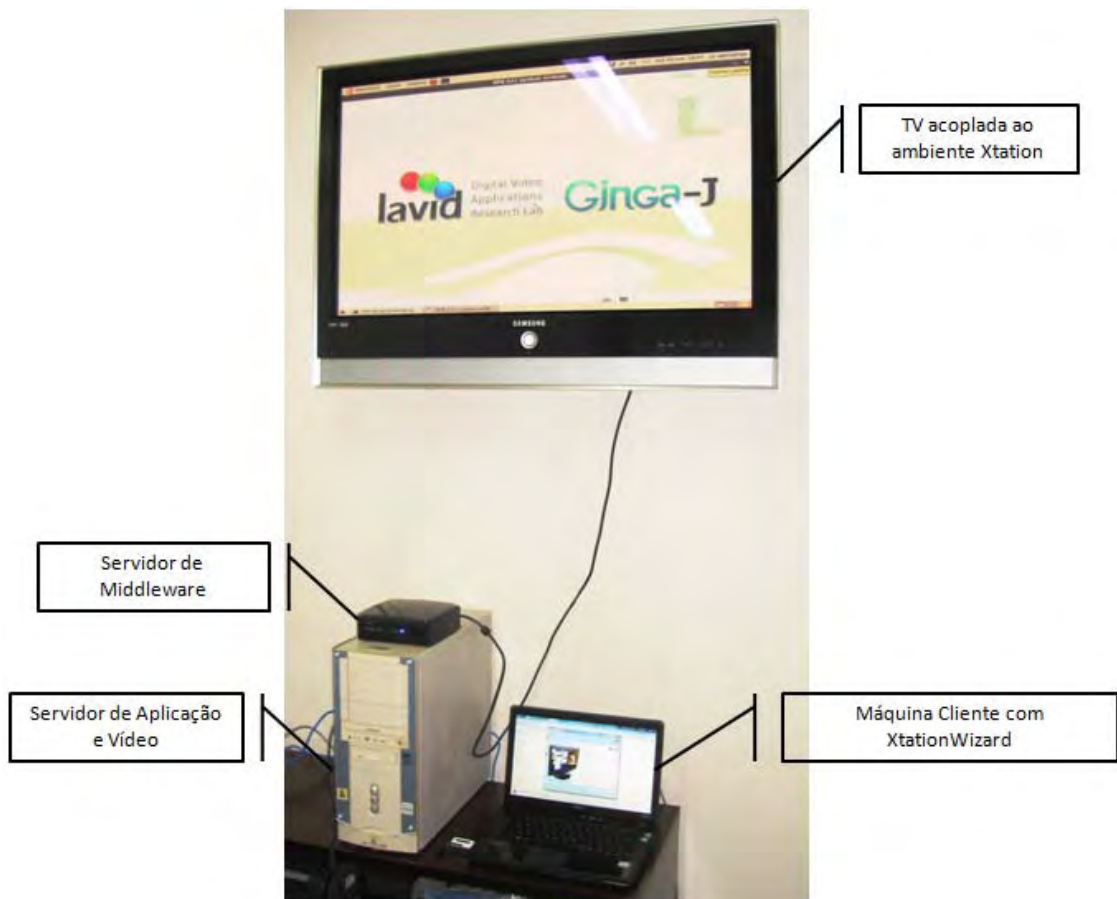


Figura 44: Infraestrutura utilizada na validação do Xtation

Tabela 11: Especificações dos equipamentos utilizados na infraestrutura do Xtation

<i>Equipamento</i>	<i>Especificação</i>
Máquina Cliente	Core 2 Duo, 2.13GHz, 3GB RAM, 300GB HD, Gigabit Ethernet. Windows 7 Home Premium. JRE 6.
Servidor de Aplicação e Vídeo	Core 2 Duo, 2.3GHz, 2GB RAM, 250GB HD, Gigabit Ethernet. Linux Debian 5.0.
Servidor de Middleware	MiniPC AOpen. Core 2 Duo, 2.16GHz, 1GB RAM, 120GB HD, interface DVI, Gigabit Ethernet. Linux Ubuntu 9.10.
TV	TV PLASMA 42". Resolução 1024x768. Interface DVI.

5.2 O PROGRAMA DE TVDI

O estudo de caso desenvolvido neste trabalho baseia-se no projeto de produção de um programa-piloto de TVDI criado pelo LAViD: a “Torcida Virtual”, ou simplesmente, “TOV”.

A Torcida Virtual (TAVARES, et al., 2004) é uma aplicação multimídia cujo principal atrativo é permitir que os usuários torçam pelo seu time, preenchendo os assentos de um estádio de esportes virtualmente. Uma vez preenchido o estádio, as pessoas participantes poderão interagir entre si como se estivessem num estádio real. A idéia é dar suporte a um ambiente virtual com espaço acústico compartilhado.

Esta aplicação é fortemente acoplada ao conteúdo audiovisual transmitido, neste caso uma partida de futebol. Desta forma, a aplicação é disponibilizada durante o tempo de duração da partida, sendo acessada pelo usuário através de um ícone interativo exibido no canto da tela, responsável por disparar a aplicação. De forma a não prejudicar a visão do usuário em todos os lances do jogo, definiu-se que a interface deve ser transparente e exibida em canto inferior direito da tela, sobre o vídeo.

A TOV foi desenvolvida em linguagem Java e suas principais funcionalidades podem ser observadas no diagrama de casos de uso UML da Figura 45. Nele podemos observar que cada participante da TOV deve escolher um setor do estádio virtual para sentar, bem como um assento numerado. A localização do usuário é importante para limitar com quais outros usuários a interação por meio do áudio se dará. Na especificação da TOV foi definido que o áudio será ouvido pelos usuários sentados próximos, virtualmente, implementando uma espécie de cone acústico. Desta forma, teremos uma interatividade permanente.

Através do caso de uso “Torcer”, o telespectador conectado pode então gerar seu áudio que será mixado e transmitido para ele e seus vizinhos do estádio. Para que essa funcionalidade se torne factível, é necessário ter algum mecanismo de captura do áudio do ambiente onde o receptor de TVD está inserido (caso de uso “Enviar áudio por dispositivo”). Neste cenário de uso, a TOV utiliza um celular conectado ao receptor para suprir esta funcionalidade.

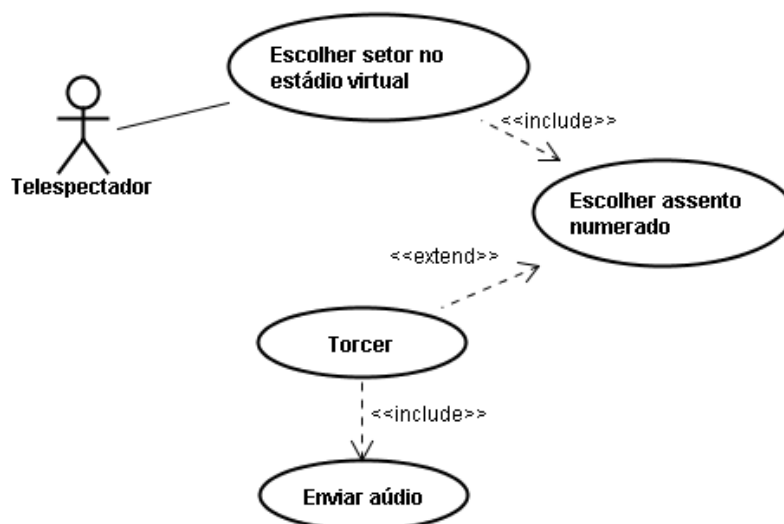


Figura 45: Casos de Uso UML para a Torcida Virtual

Vale salientar que o objetivo deste estudo de caso não é detalhar todo o processo de produção da TOV em si, mas mostrar como o Xtation pode ser útil para facilitar algumas das principais etapas deste processo: Programação Visual; Codificação do Software; Integração Software/Programa de TV; e fase de Testes, conforme detalhado no tópico a seguir.

5.3 UTILIZAÇÃO DO XTATION NO PROCESSO DE PRODUÇÃO DA TORCIDA VIRTUAL

Após a finalização de toda a fase de Concepção da TOV, a Equipe de Projeto Gráfico, responsável pelo projeto visual da interatividade do programa, está pronta para iniciar suas atividades de “Programação Visual”. Como artefatos gerados nesta etapa, teremos os protótipos do design da aplicação que deverão ser testados e validados pela equipe, até a aprovação do protótipo final. Neste momento, utilizamos o Xtation para realização do “Teste de Avaliação da Interface Gráfica”, como mostra a Figura 46. Uma aplicação simples disparada pelo XtationWizard é responsável por exibir diferentes imagens na televisão, de forma a subsidiar a escolha da interface definitiva.



Figura 46: Estudo de Caso 1: Utilização do Xtation no auxílio à atividade de Programação Visual

Após a definição da interface final, os componentes gráficos (imagens, botões, ícones, etc) gerados pela Equipe de Projeto Gráfico são enviados para a Equipe Técnica de Software, para que seja utilizada na implementação da TOV. Nesta atividade de “Codificação do Software”, o código Java desenvolvido implementa todas as telas da aplicação e suas funcionalidades e mapeia todos os eventos gerados pelo controle remoto, de acordo com as APIs especificadas no *middleware*. Através do Xtation é possível realizar “Testes de Funcionalidade” e “Testes de Interface”, onde são simulados todos os cenários de uso da TOV e seus requisitos funcionais, bem como são testadas as transições de telas e todas as interfaces existentes no sistema (Figura 47).



Figura 47: Estudo de Caso 2: Utilização do Xtation no auxílio à atividade de Codificação do Software

Paralelamente à atividade de “Codificação de Software”, a Equipe de Produção deve trabalhar na codificação das mídias que compõem o programa de TV. Estas mídias codificadas devem ser enviadas à Equipe Técnica de Software para compor o programa de TVDI, juntamente com a aplicação interativa.

No caso da TOV, utilizamos um vídeo de uma partida de futebol para simular o programa de TV em exibição. Como esta aplicação é fortemente acoplada ao conteúdo audiovisual transmitido, a aplicação deve estar disponível durante o tempo de duração da partida, devendo ser iniciada juntamente com a mesma e acessada pelo usuário através de um ícone interativo exibido no canto da tela, responsável por disparar a aplicação. É neste momento que será iniciada a atividade de “Integração Software/Programa de TV”, onde “Testes de Integração” irão verificar se juntos estes componentes funcionam corretamente, conforme o ciclo de vida especificado para cada um (Figura 48).



Figura 48: Estudo de Caso 3: Utilização do Xtation no auxílio à atividade de Integração Software/Programa TV

Com a aplicação e vídeo integrados, a Equipe Técnica de Software pode iniciar as demais atividades de “Testes”. Esta etapa abrange: “Testes de Sistema”, “Testes de Aceitação”, “Testes de Desempenho” e “Testes de Usabilidade”. Através da utilização do ambiente Xtation, podemos verificar se o produto final está de acordo com as expectativas do cliente; se aplicação possui um bom desempenho após constantes requisições ao serviço de interatividade; bem como avaliar sua atratividade e facilidade de uso, antes de o produto ser amplamente divulgado aos usuários finais. Para a realização destes testes é interessante

também verificar a acessibilidade ao programa interativo, através da utilização do controle remoto como meio de interação com o aplicativo (Figura 49). Não foi possível testar a utilização do celular como dispositivo de interação e de comunicação com o receptor para transmissão do áudio da TOV, devido às limitações da versão do *middleware* utilizado e falta de suporte a estas funcionalidades.



Figura 49: Estudo de Caso 4: Utilização do Xtation no auxílio à atividade de Testes

Um grande aliado no acompanhamento dos testes realizados é o *log* de execução. Através dele, a Equipe Técnica de Software consegue visualizar o que acontece por trás da execução da aplicação no *middleware*, recebendo informações sobre eventos disparados pelo controle remoto e pelo Xlet, detalhamento dos componentes gráficos, possíveis erros ocorridos, etc.

A Figura 50 exibe um trecho *log* de execução da TOV. Destacamos neste log algumas informações que podem ser úteis ao desenvolvedor na execução e teste de seus programas: (1) eventos originados do Gerenciador de Aplicações, informando se as funções de carregamento do Xlet foram corretamente executadas (*initXlet* e *startXlet*); (2) informação sobre a resolução da tela (*DTVContainer*), auxiliando na organização e posicionamento dos componentes gráficos; (3) informações sobre os botões pressionados, detalhando se os eventos do controle remoto foram corretamente disparados; (4) informações sobre o carregamento das imagens na tela, útil para verificar se as imagens podem ser corretamente

carregadas ou se houve algum tipo de erro durante o processo (erro no caminho da imagem, erro de formato suportado, etc).

```

Xtation - LOG
INFO->loading Xlet... [br.ufpb.lavid.ginga.xlets.tov.gui.TOVXlet]
initXlet
startXlet
INFO->XLET started... [br.ufpb.lavid.ginga.xlets.tov.gui.TOVXlet]
Resolução do DTVContainer: com.sun.dtv.lwuit.geom.Dimension[width=1024, height=768]
Carregando imagem: resources/Logo_Small.png
##### PRINT CONTAINER----->>>>> DTVContainer[x=0 y=0 width=1024 height=768]
##### PRINT COMPONENT ----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT CONTAINER----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=667 height=647]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=667 height=647]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=667 height=86]
##### PRINT COMPONENT ----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT CONTAINER----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT COMPONENT ----->>>>> Container[x=0 y=0 width=0 height=0]
##### PRINT CONTAINER----->>>>> Container[x=0 y=0 width=0 height=0]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=150 height=150]
Botão pressionado: VK_ENTER
-----Iniciando Tela Menu-----
Carregando imagem: resources/tela1.png
Carregando imagem: resources/BolaRED_OK.png
Carregando imagem: resources/Torcida_Virtual.png
Carregando imagem: resources/opcaoIndisponivel.png
##### PRINT CONTAINER----->>>>> DTVContainer[x=0 y=0 width=1024 height=768]
##### PRINT COMPONENT ----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT CONTAINER----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT CONTAINER----->>>>> Form[x=0 y=0 width=1024 height=768]
##### PRINT COMPONENT ----->>>>> Container[x=0 y=0 width=0 height=0]
##### PRINT CONTAINER----->>>>> Container[x=0 y=0 width=0 height=0]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=667 height=647]
##### PRINT COMPONENT ----->>>>> Label[x=0 y=0 width=667 height=647]

```

Figura 50: Log de execução da Torcida Virtual

Vale salientar que, em todo o estudo de caso detalhado acima, o XtationWizard foi o responsável por enviar o programa interativo (aplicação e vídeo) para o “Servidor de Aplicação e Vídeo”, que por sua vez, utilizando as ferramentas Streamer, Gerador de Carrossel e Remux, enviou o fluxo multiplexado ao “Servidor de Middleware” onde encontra-se instalado o OpenGinga. O OpenGinga recebe este fluxo e exibe-o na TV, além de ser o responsável pelo correto funcionamento da aplicação. Desta forma, temos uma simulação de um ambiente de TVDI real, auxiliando todo o processo de produção de programas interativos.

6

CONSIDERAÇÕES FINAIS

O ambiente descrito nesta dissertação tem características que o tornam bastante útil ao processo de produção de programas de TVDI. Como este processo envolve uma equipe multidisciplinar, é necessário que tenhamos ferramentas simples e que garantam simulações próximas ao ambiente real de forma a validar o programa interativo antes de sua veiculação aos telespectadores.

Sem ferramentas como a apresentada aqui, os designers não conseguiriam ter a visão real da apresentação de seu trabalho em uma plataforma de TVDI e nem os desenvolvedores conseguem executar e testar suas aplicações de forma satisfatória, principalmente devido à grande dependência com a especificação do *middleware* utilizada e à necessidade de ferramentas de transmissão e recepção do fluxo multiplexado. Além disso, os produtores e clientes também não conseguiriam ter uma visão clara do produto final desenvolvido antes de sua liberação.

Outro ponto interessante é que o Xtation também atua como uma forma de divulgação do *middleware* Ginga, permitindo o crescimento da demanda por programas de TVDI pela comunidade e para ela.

Contudo, apesar dos bons resultados obtidos, discutiremos adiante as principais dificuldades encontradas na realização deste trabalho, bem como as oportunidades que podem ser aproveitadas em trabalhos futuros.

6.1 HISTÓRICO E DIFICULDADES ENCONTRADAS

Um primeiro esboço deste trabalho foi demonstrado, com sucesso, no Salão de Ferramentas do WebMidia (SILVA, et al., 2007). Desde a primeira transmissão oficial de TV Digital no Brasil, que ocorreu no final de 2007, novos desafios surgiram com as discussões sobre o *middleware* utilizado no SBTVD, que culminou com a especificação do Ginga, e com a reformulação da arquitetura de sua versão procedural. A TV Digital já é realidade na maioria das capitais brasileiras, porém esta TV ainda não é interativa. Atrasos na definição do *middleware* Ginga justificam a timidez com que os receptores com suporte a interatividade e as empresas que desenvolvem aplicativos interativos estão entrando no mercado.

O fato é que sem *middleware* não há aplicações interativas, o que impacta diretamente no desenvolvimento de ferramentas de apoio ao processo de produção de conteúdo para TVDI, a exemplo do Xtation. Uma vez definida a especificação Ginga, o mercado de software para TVDI começa a lançar implementações do *middleware* brasileiro. Este mercado exige produtos alinhados com os anseios do grande público que aguarda ansiosamente a interatividade.

Um dos problemas enfrentados durante o desenvolvimento deste projeto e que merece destaque diz respeito à reformulação da arquitetura do Ginga-J e, conseqüentemente, do OpenGinga. O Ginga-J utilizava as APIs do GEM e, portanto, poderia ser alvo da cobrança de *royalties*. A discussão acerca deste tema por parte do Fórum do Sistema Brasileiro de TV Terrestre²⁸ atrasou o lançamento da parte procedural do *middleware* Ginga, o que também inibiu os avanços no desenvolvimento dos programas interativos. Para solucionar este problema, o desenvolvimento de uma especificação aberta e *royalties-free* foi iniciado, esforço de uma atividade conjunta do SBTVD, do governo brasileiro e da Sun Microsystems. No final de 2008, a Sun Microsystems lançou a primeira *release* do JavaDTV, plataforma aberta, interoperável e *royalty-free* que permite a implementação de serviços interativos com Java através do Ginga-J.

²⁸ Conselho formado por membros da academia e do setor privado que discutem a padronização do Sistema Brasileiro de TV Digital (SBTVD) - <http://www.forumsbtvd.org.br/>

Finalmente, em meados de 2009, o Fórum SBTVD anunciou o que o JavaDTV seria utilizado pelo Ginga-J. Como se trata apenas de uma especificação, cabe aos desenvolvedores de *middleware* a efetiva implementação do conjunto padronizado de classes. Uma nova versão do OpenGinga contemplando esta nova arquitetura ainda está em desenvolvimento, o que causou atraso na finalização deste projeto.

É fato que qualquer ambiente de execução e testes de programas de TVDI tem dependência direta da utilização de um *middleware* que dê suporte ao mesmo e implemente todas as funcionalidades necessárias à interatividade e à correta exibição de vídeo e aplicações. Contudo, até o momento do término deste trabalho, não foi lançada uma versão totalmente estável do OpenGinga sendo este o principal problema enfrentado para a finalização do Xtation e validação do estudo de caso.

Há limitações quanto à máquina virtual Java utilizada (as aplicações devem ser compiladas para Java 1.4); as APIs da especificação Ginga-J ainda não estão totalmente implementadas, o que restringe bastante as funcionalidades da aplicação; a implementação do gerenciador de aplicações (*Application Manager*) não está satisfatória, gerando demora no carregamento e execução da aplicação interativa; e há também limitações na execução da API multiusuários e multidispositivos o que dificulta, inclusive, a utilização do celular na “Torcida Virtual”.

6.2 OPORTUNIDADES

Conforme detalhado nos capítulos anteriores, o escopo deste trabalho refere-se à utilização do Xtation como uma ferramenta que viabilize e facilite o processo de produção de programas para TVDI. No entanto, novas oportunidades podem ser vislumbradas, a exemplo da utilização deste ambiente para a realização de testes de conformidade e integração dos componentes e das implementações do *middleware* Ginga.

Para tanto, o Xtation seria responsável por coletar e armazenar resultados de testes do *middleware*, realizar tratamento do *log* de execução e analisar testes individuais e comparações de testes de desempenho. Isto se tornaria possível através de um *framework*

de aplicações contendo suítes de testes a serem executadas através do Xtation. Estas aplicações carregariam testes de desempenho gráfico, persistência, segurança, utilização de canal de retorno, comunicação entre aplicações, entre outros, de forma a validar as implementações dos componentes de acordo com a especificação Ginga.

Outra possibilidade a ser trabalhada diz respeito à utilização do Xtation para realização de testes de categorias de aplicações interativas, com base na definição de uma taxonomia que leve em consideração características como: natureza das aplicações, finalidade das aplicações, perfil dos usuários potenciais, mecanismos e dispositivos de interação, entre outras. Desta forma, podem ser definidos casos de teste específicos para cada categoria, sendo o Xtation responsável por viabilizar a execução dos mesmos.

Além disto, o Xtation pode ser utilizado como parte integrante de uma infraestrutura que permita a realização de testes remotos de programas interativos, através de uma interface *web* que permita *upload* de arquivos e de componentes que permitam a visualização do programa executado no *middleware* e a manipulação remota através de controle remoto virtual. Há ainda a possibilidade de utilização do ambiente Xtation como ferramenta que permita o teste de aplicações de TVDI em diferentes implementações de *middleware*, comparando o comportamento da aplicação nas diferentes plataformas.

7

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT NBR 15606. 2009. *Televisão digital terrestre — Codificação de dados e especificações de transmissão para radiodifusão digital — Parte 4: Gíngam-J — Ambiente para a execução de aplicações procedurais.* 2009.

ADAMS, M., ANAND, P. e FOX, S. 2001. *Interactive television: coming soon to a television near you.* 2001.

ALVES, K. C., et al. 2009. Uma proposta de telejornal educativo interativo. [A. do livro] Pedro (org) NUNES. *Mídias Digitais & Interatividade.* João Pessoa : Editora UFPB, 2009.

ANATEL – Agência Nacional de Telecomunicações. 2001. *Relatório Integrador dos Aspectos Técnicos e Mercadológicos da TV Digital.* 2001.

ArviD. 2004. *A Guide to the OpenMHP environment.* Helsinki : s.n., 2004.

BARBOSA NETO, Armando. 2008. *TV DIGITAL INTERATIVA: Os novos rumos da produção televisiva.* 2008.

BARROS, Gil Garcia de. 2006. *A consistência da interface com o usuário para a TV interativa.* São Paulo : s.n., 2006. Dissertação de Mestrado apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

BATISTA, C. E. C. F. 2007. *TVGrid - Computação em grid em uma Rede de TV Digital.* Departamento de Informática, Universidade Federal da Paraíba. 2007. Dissertação (mestrado).

BECK, Kent, et al. 2001. Manifesto for Agile Software Development. [Online] 2001.
<http://www.agilemanifesto.org/>.

BECKER, Valdecir. 2006. *Concepção e Desenvolvimento de Aplicações Interativas para Televisão Digital*. Universidade Federal de Santa Catarina. Florianópolis : s.n., 2006. Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia e Gestão do Conhecimento.

BRASIL. 2006. *Decreto No 5.820, de 29 de Junho de 2006*. 2006. Dispõe sobre a implantação do SBTVD-T, estabelece diretrizes para a transição do sistema de transmissão analógica para o sistema de transmissão digital do serviço de radiodifusão de sons e imagens e do serviço de retransmissão de televisão.

CAREY, John. 1996. *Winky Dink To Stargazer: Five Decades of Interactive Television. Responsibility In The New Media Landscape*. Amsterdam : European Institute For The Media, 1996.

CHORIANOPOULOS, Konstantinos. 2004. *Virtual Television Channels: Conceptual Model, User Interface Design and Affective Usability Evaluation*. Department of Management Science and Technology Athens , University of Economics and Business. 2004. p. 180, Thesis submitted for the degree of Doctor of Philosophy.

CPqD. 2006. Modelo de referência: Sistema Brasileiro de Televisão Digital Terrestre. 2006.

DEITEL, H. M. e DEITEL, P. J. 2003. *Java: como programar*. Porto Alegre : Bookman, 2003.

GAWLINSKI, Mark. 2003. *Interactive television production*. Oxford : Focal Press, 2003.

GERTZEN, C. 2005. *User Guideline for the IRT-Testcentre*. Munich : s.n., 2005.

GUEDES, Gilleanes. 2006. *UML Uma Abordagem Prática*. 2a Edição. São Paulo : Editora Novatec, 2006.

IRT. 2005. *IRT's MHP Test Centre*. Munich : s.n., 2005.

ITU. 2001. *ITU-T Recommendation J.200: Worldwide common core – Application environment for digital interactive television services*. 2001.

—. **2004.** *ITU-T Recommendation J.201: Harmonization of declarative content format for interactive television applications*. 2004.

—. **2003.** *ITU-T Recommendation J.202: Harmonization of procedural content formats for interactive TV applications*. 2003.

JACOBSON, Ivar, RUMBAUGH, James e BOOCH, Grady. 2005. *UML Guia do Usuário*. 2a Edição. Rio de Janeiro : Editora Campus, 2005.

JENSEN, Jens F. 2007. *About Interactive Television*. 2007.

- LAMONT, Sheri. 2003.** An 8-step process for creating compelling enhanced television. *Proceedings of the 1st European Conference on Interactive Television: from Viewers to Actors?* 2003.
- LARMAN, Craig. 2002.** *Utilizando UML e Padrões: uma introdução à Análise e ao Projeto.* 2ª edição. Porto Alegre : Bookman, 2002.
- LEITE, L. E. C., et al. 2005.** FlexTV – Uma Proposta de Arquitetura de Middleware para o Sistema Brasileiro de TV Digital. *Revista de Engenharia de Computação e Sistemas Digitais.* 2005, Vol. 2, pp. 29-50.
- LU, Karyn Y. 2005.** *Interaction Design Principles for Interactive Television.* Georgia Institute of Technology. 2005. p. 202, Thesis Presented to The Academic Faculty Degree Master of Science in Information Design and Technology.
- LUNDSTROM, Lars-Ingemar. 2006.** *Understanding Digital Television - An Introduction to DVB Systems with Satellite, Cable, Broadband, and Terrestrial TV Distribution.* s.l. : Focal Press, 2006.
- MÉDOLA, Ana Silvia Lopes Davi e TEIXEIRA, Lauro Henrique de Paiva. 2007.** Aspectos da TV Digital interativa: como pode ficar a nova televisão do ponto de vista do usuário. [A. do livro] Clodoaldo M. (Org.) Cardoso. *Diversidade e igualdade na comunicação - coletânea de textos do Fórum da Diversidade e Igualdade: cultura, educação e mídia.* Bauru : s.n., 2007.
- MHP-KDB. 2004.** *The MHP Knowledge Project - Definition of Overall Project Architecture.* Munich : s.n., 2004.
- MONTEZ, Carlos e BECKER, Valdecir. 2005.** *TV Digital Interativa: conceitos, desafios e perspectivas para o Brasil.* 2.ed. ver. e ampl. Florianópolis : Editora UFSC, 2005.
- MORRIS, S. e SMITH-CHAIGNEAU, A. 2005.** *Interactive TV Standards – A Guide to MHP, OCAP and JavaTV.* s.l. : Focal Press, 2005.
- MORRIS, Steven. 2008.** <http://www.interactivetvweb.org/>. *TV Without Borders.* [Online] 2008.
- PRESSMAN, Roger S. 2002.** *Engenharia de Software.* 5.ed. Rio de Janeiro : Mc Graw-Hill, 2002.
- RIBEIRO, Nuno. 2004.** *Tecnologia de Informação: Multimédia e Tecnologias Interativas.* Lisboa : FCA Editora de Informática, 2004.
- SCHWALB, Edward M. 2003.** *iTV Handbook: Technologies and Standards.* s.l. : Prentice Hall PRT, 2003.
- SILVA, F. P. R., et al. 2007.** Xtation: um Ambiente de Testes de Aplicações para TV Digital Interativa Baseado no Middleware de Referência do Sistema Brasileiro de Televisão Digital. *Brazilian Symposium on Multimedia and the Web - WebMedia.* 2007.
- SILVA, Lincoln David Nery. 2008.** *Uma Proposta de API para Desenvolvimento de Aplicações Multiusuário e Multidispositivo para TV Digital Utilizando o Middleware Ginga.* Departamento de Informática, Universidade Federal da Paraíba. 2008. p. 77, Dissertação (mestrado).

- SILVA, Lincoln David Nery, et al. 2007.** Suporte para desenvolvimento de aplicações multiusuário e multidispositivo para TV Digital com Ginga. *T&C Amazônia Magazine*. N. 12, 2007, pp. 75-84.
- SILVA, Marco. 1998.** O que é interatividade? *Boletim Técnico do SENAC*. [Online] 1998. <http://www.senac.br/informativo/bts/242/boltec242d.htm>.
- SOARES, L. F. G. 2006.** MAESTRO: The Declarative Middleware Proposal for the SBTVD. *Proceedings of the 4th European Interactive TV Conference*. 2006.
- SOARES, Luiz Fernando Gomes. 2007.** Ambiente para Desenvolvimento de aplicações Declarativas para a TV Digital Brasileira. *MDIC*. 2007.
- SOARES, Luiz Fernando Gomes, RODRIGUES, Rogério Ferreira e MORENO, Márcio Ferreira. 2007.** Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. *Journal of the Brazilian Computer Society*. 2007, Vol. v12, pp. 37-46.
- SOARES, Michel S. 2004.** Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. *Journal of Computer Science*. 2004, Vol. v3, pp. 8-13.
- SOMMERVILLE, Ian. 2004.** *Software Engineering*. 7th edition. s.l. : Pearson Education, 2004.
- SOUTO MAIOR, Marcelo. 2002.** *A TV interativa e seus caminhos*. Instituto de Computação - Universidade Estadual de Campinas. Campinas : s.n., 2002. p. 72, Dissertação Final de Mestrado Profissional (Mestrado em Computação).
- SOUZA FILHO, G. L. de, LEITE, L. E. C. e BATISTA, C. E. C. F. 2007.** Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. *Journal of the Brazilian Computer Society*. 2007, Vol. v12, pp. 47-56.
- SOUZA, Cidcley T. de e OLIVEIRA, Carina T. de. 2005.** Especificação de Canal de Retorno em Aplicações para TV Digital Interativa. *XXII Simpósio Brasileiro de Telecomunicações*. 2005.
- STRAUBHAAR, Josef e LAROSE, Robert. 2004.** *Comunicação, mídia e tecnologia*. São Paulo : Pioneira Thompson Learning, 2004.
- SUN. 2008.** Java DTV API 1.0 Specification. s.l. : Sun Microsystems, 2008.
- TAVARES, T. A., et al. 2004.** Sharing Virtual Acoustic Spaces over Interactive TV Programs: Presenting Virtual Cheering Application. *International Conference on Multimedia and Expo (ICME)*. 2004.
- TAVARES, Tatiana Aires. 2004.** *Proposta de uma Metodologia para o Desenvolvimento de Programas de TV Interativa baseada em Métodos Ágeis para o Desenvolvimento de Software*. 2004. Relatório Técnico (Projeto HiTV).
- TEIXEIRA, Lauro Henrique de Paiva e CASELLA, César Fernandes. 2008.** Televisão Digital Interativa: A Usabilidade como Linguagem de Uso. *Revista do NP em Comunicação Audiovisual da Intercom*. 2008, Vol. I, 211-228.

REFERÊNCIAS BIBLIOGRÁFICAS

VEIGA, Elba Guimarães e TAVARES, Tatiana Aires. 2007. Um Modelo de Processo para o Desenvolvimento de Programas para TV Digital e Interativa baseado em Metodologias Ágeis. 2007.

VEIGA, Elba Guimarães. 2006. *Modelo de Processo de Desenvolvimento de Programas para TV Digital e Interativa.* Universidade Salvador. 2006. Dissertação (mestrado).

WAISMAN, Thais. 2006. *Usabilidade em serviços educacionais em ambiente de TV Digital.* São Paulo : s.n., 2006. Tese (doutorado) - Escola de Comunicação e Artes da Universidade de São Paulo.

WHITAKER, Jerry. 2001. *Interactive Television Demystified.* s.l. : McGraw-Hill, 2001.