

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

FRANCO GARCIA BORTOLOTTO

**ESTUDO SOBRE A TELEVISÃO DIGITAL E DESENVOLVIMENTO DE UM
PROTÓTIPO INTERATIVO BASEADO NO SUBSISTEMA PROCEDURAL DO
MIDDLEWARE GINGA**

CRICIÚMA, DEZEMBRO DE 2010

FRANCO GARCIA BORTOLOTTO

**ESTUDO SOBRE A TELEVISÃO DIGITAL E DESENVOLVIMENTO DE UM
PROTÓTIPO INTERATIVO BASEADO NO SUBSISTEMA PROCEDURAL DO
MIDDLEWARE GINGA**

Trabalho de Conclusão apresentado para
obtenção do Grau de Bacharel em Ciência da
Computação da Universidade do Extremo Sul
Catarinense.

Orientador: Prof. MSc. Paulo João Martins

Co-Orientador: Prof. Esp. Fabricio Giordani

CRICIÚMA, DEZEMBRO DE 2010

FRANCO GARCIA BORTOLOTTO

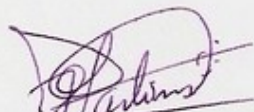
Estudo sobre a Televisão Digital e Desenvolvimento de um Protótipo Interativo Baseado no Subsistema Procedural do *Middleware* Ginga

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.




Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

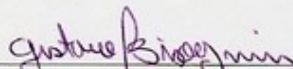
Banca Examinadora:



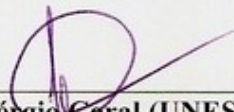
Prof. MSc. Paulo João Martins (UNESC)
Orientador



Prof. Esp. Fabrício Giordani (UNESC)
Co-Orientador



Prof. MSc. Gustavo Bisognin (UNESC)



Prof. Esp. Sérgio Coral (UNESC)

Aos meus ex-colegas e amigos do Setor de Informática da Damyller Jeans, pelos momentos alegres lá vividos, pelos momentos de angústia, tensão e “plantões” infinitos compartilhados e por tudo o que aprendemos juntos ao longo de seis anos.

AGRADECIMENTOS

Aos professores do curso de Ciência da Computação, pelas experiências compartilhadas ao longo destes anos, estendendo também àqueles que não mais fazem parte do corpo docente.

Ao Paulo (orientador) e Fabrício (co-orientador), pelo tempo e paciência dispensados, sugestões e contribuições durante a elaboração deste projeto.

À coordenação e secretaria do curso, pelo atendimento gentil e eficiente.

À minha família, amigos e colegas de trabalho, pela compreensão dispensada nos momentos cruciais e estressantes e por suas contribuições, diretas ou indiretas.

Aos colegas e ex-colegas de curso, pelos momentos vividos, trocas de ideias e discussões – por vezes de forma calorosa – durante este trajeto.

À Faculdade de Tecnologia em Automação Industrial da SATC. Embora não tenha concluído o curso iniciado no ano 2000, agradeço pela base acadêmica lá adquirida.

À empresa Damyller Jeans e ex-colegas de trabalho no Setor de Informática, por todo o conhecimento e experiência adquiridos ao longo dos nove anos em que fiz parte de sua equipe de colaboradores.

Enfim, agradeço de modo especial: Sigvard von Baranow Jr., por um dia ter colocado um livro de Java em minhas mãos, pelo incentivo e pelas boas práticas de programação; Aldenir Pedro Marangoni, pelas lições de Bancos de Dados, entidades e relacionamentos; e Ana Paula Minatto Nuernberg, por tudo o que me ensinou sobre SQL, relatórios, programação e testes.

Talvez tenha omitido alguns nomes. Sendo assim, a todos muito obrigado.

”(...) A educação digital deve ser parte do planejamento do País, que integra os excluídos do mundo virtual para que não se alimente a exclusão social”.

(Gustavo Henrique Carvalho Schiefler)

RESUMO

A Televisão Digital aberta vem ganhando destaque nos países emergentes. Além de oferecer som e imagem em qualidade e definição superiores, a digitalização do sinal deflagra uma revolução no sistema televisivo, disponibilizando via difusão conteúdos em novos formatos e suportando o funcionamento de um mecanismo de interatividade que transforma o telespectador em usuário ativo. No Brasil, os trabalhos visando à digitalização vêm sendo realizados desde os anos 90 com base nos padrões americano, europeu e japonês. Por iniciativa do Governo Federal, que enxerga a interatividade da TV Digital como instrumento no processo de inclusão digital e social da população carente, os trabalhos foram conduzidos culminando com o lançamento do Sistema Brasileiro de Televisão Digital via decreto presidencial nº 5820, de 2006, tendo como padrão o ISDB-Tb, uma adaptação do sistema japonês incluindo inovações voltadas à realidade nacional. O *middleware* é considerado como a inteligência por trás da TV Digital interativa, suportando o mecanismo de interatividade e a execução de aplicações no receptor. A especificação de *middleware* escolhida para o SBTVD foi o Ginga, oferecendo um ambiente para desenvolvimento e execução de aplicações em paradigma declarativo, procedural ou híbridas, utilizando os recursos de hardware, software e de canais de comunicação (Internet). O protótipo desenvolvido em Java utilizando a implementação de referência OpenGinga – baseada no subsistema procedural - serve como demonstrativo do potencial oferecido pela especificação e seus subsistemas, viabilizando a construção de aplicações com interfaces elaboradas, ricas e interativas, utilizando uma rede ou a Internet como canal de retorno suportando a interatividade remota.

Palavras-chave: TV Digital Interativa; Sistema Brasileiro de Televisão Digital; Middleware Ginga; Java; OpenGinga.

ABSTRACT

Open Digital television has been gaining attention in emerging markets. In addition to providing sound and image quality and higher definition, the digitization of the signal triggers a revolution in the television system, available via broadcast contents in new formats and supporting the operation of an interactive mechanism that turns viewers into active users. In Brazil, the works towards the digitization has been made since the 90s based on American, European and Japanese standards. On the initiative of the Federal Government, which sees the interactive digital TV as a tool in the progress of digital and social inclusion of the poor, the works were conducted culminating in the launch of the Brazilian System of Digital Television via Presidential Decree No. 5820, 2006, having the ISDB-TB standard, an adaptation of the Japanese system including innovations aimed at national reality. The middleware is considered as the intelligence behind the Interactive Digital TV, supporting the mechanism of interaction and implementation of applications in the receiver. The specification of middleware has been chosen to SBTVD Ginga, offering an environment for developing and running applications in paradigm declarative, procedural or hybrid using the resources of hardware, software and communication channels (Internet). The prototype developed in Java using the reference implementation OpenGinga – based on procedural subsystem – serves as a demonstration of the potential offered by the specification and its subsystems, making possible the construction of applications with developed, rich and interactive interfaces, using a network or the Internet as a channel supporting the return of remote interactivity.

Key-words: Interactive Digital TV; Brazilian System of Digital Television; Middleware Ginga, Java, OpenGinga.

LISTA DE ILUSTRAÇÕES

Figura 1. Divisão do espectro de transmissão, possibilitando a multiprogramação.....	29
Figura 2. Visualização do sistema de TV Digital e integração entre as partes envolvidas	32
Figura 3. Organização em <i>bouquet</i> , compartilhado entre diferentes meios de transmissão.....	33
Figura 4. Arquitetura de camadas de um sistema de TV Digital e tecnologias envolvidas	35
Figura 5. Diagrama do fluxo de informação entre as camadas do modelo genérico.....	36
Figura 6. Modelo ITU-T para composição e transmissão do sinal de TV Digital	37
Figura 7. a) captura de sinais analógicos; b) amostragem e quantização do sinal	37
Figura 8. Multiplexação de áudio e vídeo em um <i>stream</i> MPEG-1 <i>System</i>	39
Figura 9. Fluxo de transporte contendo as tabelas PSI básicas e fluxos elementares	40
Figura 10. Fluxo de transporte e seus subfluxos	40
Figura 11. Modulação por amplitude, frequência ou fase	41
Figura 12. Funcionamento do carrossel de dados.....	43
Figura 13. a) Envio via carrossel; b) Tratamento de arquivo corrompido	44
Figura 14. Etapas do proceso de recepção do sinal digital	45
Figura 15. Exemplar de receptor <i>set-top box</i>	46
Figura 16. Arquitetura de referência do terminal de acesso	47
Figura 17. O <i>middleware</i> e seu papel intermediador entre as aplicações e a plataforma.....	48
Figura 18. Arquitetura do ambiente de execução conforme ITU-T J.200.....	51
Figura 19. O GEM e a relação com os <i>middlewares</i> dos padrões de TV Digital aberta.....	55
Figura 20. Diagrama de blocos do sistema ATSC	56
Figura 21. Diagrama de blocos da estrutura do sistema DVB	57
Figura 22. Diagrama de blocos do sistema ISDB.....	58
Figura 23. a) aplicação <i>enhanced tv</i> da rede CNN; b) aplicação <i>t-commerce</i>	66

Figura 24. Aplicação de guia de programação eletrônica	68
Figura 25. Diagrama simplificado do canal de interatividade.....	69
Figura 26. Arquitetura do canal de interatividade e as múltiplas tecnologias alternativas	72
Figura 27. Diagrama de blocos das sete normas/especificações do ISDB-Tb	78
Figura 28. Transmissão OFDM com banda segmentada: BST-OFDM	80
Figura 29. Configuração básica de um receptor ISDB-Tb	83
Figura 30. Camadas do aparelho receptor do SBTVD	84
Figura 31. Arquitetura de referência do <i>middleware</i> Ginga	87
Figura 32. Arquitetura em camadas do <i>middleware</i> Ginga	89
Figura 33. Arquitetura do Ginga NCL.....	90
Figura 34. Etapas de uma aplicação Java	92
Figura 35. Conjuntos de API's que compõem o subsistema Ginga-J	93
Figura 36. Xlets, gerente de aplicações e máquina virtual Java	94
Figura 37. Estrutura do Ginga,após a substituição dos pacotes GEM por JavaDTV	98
Figura 38. Estrutura de camadas para apresentação de serviços na tela do receptor	100
Figura 39. Estrutura de pacotes do <i>middleware</i> Ginga.....	101
Figura 40. Contexto de arquiteturas envolvidas no modelo de execução do Ginga-J.....	102
Figura 41. Arquitetura Ginga-J e ambiente de execução.....	103
Figura 42. Diagrama com estados do ciclo de vida de um Xlet	106
Figura 43. Arquitetura do OpenGinga <i>release</i> 1, contendo as bibliotecas Java DTV.....	108
Figura 44. Interface do protótipo de jogo educacional Quiz Edtec	109
Figura 45. Interface do protótipo Diga Doctor	110
Figura 46. Interface do protótipo de T-Commerce.....	111
Figura 47. Interface do protótipo de portal de controle de dispositivos residenciais	112
Figura 48. Interface do protótipo de aplicação baseada no subsistema Ginga-NCL.....	113

Figura 49. Evolução do desenvolvimento de acordo com o modelo Iterativo Incremental ...	116
Figura 50. Modelo Iterativo Incremental e seu ciclo de desenvolvimento progressivo	117
Figura 51. Esboço da interface da aplicação definida no Workshop hipotético.....	119
Figura 52. Diagrama de casos de uso da interface do aplicativo T-Commerce	123
Figura 53. Diagrama de casos de uso do cadastro de usuário/cliente.....	123
Figura 54. Matriz de requisitos funcionais x casos de uso	124
Figura 55. Método <i>processRequest</i> , direcionando as requisições no servidor	125
Figura 56. Emprego da plataforma CDC/FP/PBP 1.1 na IDE Eclipse.....	128
Figura 57. Trecho do método <i>initXlet</i> -captura do contexto e configuração do dispositivo...	130
Figura 58. Criação do componente gráfico principal e <i>widgets</i> contidos por ele.....	131
Figura 59. <i>startXlet</i> : captura do contêiner, adição do formulário e exibição da interface.....	132
Figura 60. Tela inicial do protótipo T-Commerce, com as instruções básicas de uso	133
Figura 61. Inicialização das variáveis de controle (chaves de estado) da aplicação	134
Figura 62. Requisição ao servidor e recebimento da lista de objetos serializados.....	135
Figura 63. Tratamento da requisição <i>retornaCategoria</i> pelo servidor, retornando objetos ...	136
Figura 64. Implementação de <i>UserInputEventListener</i> e sobrescrita do ouvinte.....	137
Figura 65. Exibição de <i>Calçados</i> no painel inferior, e detalhes de um item selecionado	138
Figura 66. Trecho da classe <i>Cryptor</i> , onde se vê a chave de 256 bits e a encriptação	139
Figura 67. Cifragem/envio de objeto no servidor e recepção/decifragem no cliente	140
Figura 68. Tela de finalização/encerramento do processo de compra.....	141
Figura 69. a) Envio de parâmetros sem criptografia; b) envio após criptografia	144
Figura 70. Objeto <i>Cliente</i> cifrado no <i>WebService</i> , antes de ser enviado ao cliente.....	145

LISTA DE TABELAS

Tabela 1. Principais diferenças entre TV analógica e TV Digital	31
Tabela 2. Comparativo entre os <i>middlewares</i> MHP, DASE e ARIB	53
Tabela 3. Comparativo entre os padrões americano, europeu e japonês	58
Tabela 4. Classificação dos níveis de interatividade em função do canal de retorno.....	64
Tabela 5. Proporção de domicílios com acesso à Internet no Brasil	70
Tabela 6. Descrição das especificações do ISDB-Tb e respectivas normas ABNT	79
Tabela 7. Perfis do <i>codec</i> de áudio MPEG-4 HE-AAC	81
Tabela 8. Perfis do <i>codec</i> de vídeo MPEG-4 H.264.....	82
Tabela 9. Lista de Requisitos Funcionais levantados (continua).....	120
Tabela 9. Lista de Requisitos Funcionais levantados (continuação).....	121
Tabela 10. Lista de Requisitos Não Funcionais levantados	121
Tabela 11. Listagem de casos de uso em forma resumida.....	122

LISTA DE SIGLAS

ACAP	<i>Advanced Common Application Platform</i>
8-VSB	<i>8 Vestigial Side Band</i>
A/V	<i>Áudio/Vídeo</i>
ABERT	<i>Associação Brasileira de Emissoras de Rádio e TV</i>
ABNT	<i>Associação Brasileira de Normas Técnicas</i>
AC3	<i>Audio Compression 3</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
ANATEL	<i>Agência Nacional de Telecomunicações</i>
API	<i>Application Programming Interface</i>
ARIB	<i>Association of Radio Industries and Businesses</i>
ATSC	<i>Advanced Television Systems Committee, Inc</i>
BBC	<i>British Broadcast Corporation</i>
BLOG	<i>Web Log</i>
BML	<i>Broadcast Markup Language</i>
BP	<i>Baseline Profile</i>
BST-OFDM	<i>Band-Segmented Transmission Orthogonal Frequency Division Multiplexing</i>
CD	<i>Compact Disc</i>
CDC	<i>Java Connected Device Configuration framework</i>
CDMA	<i>Code Division Multiple Access</i>
CETICbr	<i>Centro de Estudos sobre as Tecnologias da Informação/Comunicação</i>
CIF	<i>Common Intermediate Format</i>
COFDM	<i>Coded Orthogonal Frequency Division</i>

Com-Tv	Comissão Assessoria para Assuntos de Televisão
CPqD	Centro de Pesquisa e Desenvolvimento em Telecomunicações
CPU	<i>Central Process Unit</i>
CSS	<i>Cascading Style Sheets</i>
DAO	<i>Data Access Object</i>
DASE	<i>DTV Application Software Environment</i>
DAVIC	<i>Digital Audio Visual Council</i>
DDB	<i>Download Data Block</i>
DSM-CC	<i>Digital Storage Media, Command and Control</i>
DTV	<i>Digital TeleVision</i>
DVB	<i>Digital Video Broadcasting</i>
DVB-HTML	<i>Digital Video Broadcasting - HyperText Markup Language</i>
DVB-J	<i>Digital Video Broadcasting - Java</i>
DVD	<i>Digital Video Disc ou Digital Versatile Disc</i>
EDTV	<i>Enhanced Definition Television</i>
EPG	<i>Electronic Program Guide</i>
FP	<i>Java Foundation Profile</i>
GEM	<i>Globally Executable MHP</i>
GIF	<i>Graphics Interchange Format</i>
GINGA CDN	<i>Ginga Code Development Network</i>
GINGA-CC	<i>Subsistema Ginga Common-Core</i>
GINGA -J	<i>Subsistema Ginga Java</i>
GINGA -NCL	<i>Subsistema Ginga Nested Context Language</i>
GNU	<i>Gnu's Not Unix = General Public License</i>
GPLv2	<i>GNU General Public License, version 2</i>

GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communication</i>
HAN	<i>Home Area Network</i>
HAVi	<i>Home Audio Video</i>
HD DVD	<i>High-Definition/Density DVD</i>
HDCP	<i>High-bandwidth Digital Content Protection (Intel)</i>
HDTV	<i>High Definition Television</i>
HP	<i>High Profile</i>
HTML	<i>HiperText Markup Language</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
ISDB-Tb	<i>Brazilian Integrated Services Digital Broadcasting Terrestrial</i>
ISO	<i>International Organization for Standardization</i>
ITU	<i>International Telecommunication Union</i>
ITU-T	<i>ITU: Telecommunication Standardization Sector</i>
J2ME	<i>Java 2 Micro Edition Platform</i>
JAR	<i>Java Archive</i>
JMF	<i>Java Media Framework</i>
JPEG	<i>Joint Photographic Experts Group</i>
JVM	<i>Java Virtual Machine</i>
LAViD	<i>Laboratório de Aplicações de Vídeo Digital</i>
LDTV	<i>Low Definition Television</i>
LWUIT	<i>Lightweight User Interface Toolkit</i>

MHP	<i>Multimedia Home Platform</i>
MP3	<i>MPEG-1 Audio Layer 3</i>
MPEG	<i>Moving Picture Experts Group</i>
NBR	Norma Brasileira (nomenclatura as para normas do conjunto ABNT)
NCL	<i>Nested-Context Language</i>
NCM	<i>Nested-Context Model</i>
NHK	<i>Nippon Hōsō Kyōkai</i> (corporação de radiodifusão do Japão)
OCAP	<i>OpenCable Applications Platform</i>
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
OS	<i>Operational System</i>
OSI	<i>Open Systems Interconnection</i>
PAT	<i>Program Association Table</i>
PBF	<i>Java Personal Basis Profile</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
PES	<i>Packetized Elementary Stream</i>
PID	<i>Process Identifier</i>
PLC	<i>Power Line Communication</i>
PUC	Pontificia Universidade Católica
PVR	<i>Personal Video Recorder</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
QVGA	<i>Quarter Video Graphics Array</i>
ROM	<i>Read-Only Memory</i>
RS-232	<i>Recommended Standard 232</i>

SBTVD	Sistema Brasileiro de Televisão Digital
SBTVD-T	Sistema Brasileiro de Televisão Digital Terrestre
SCMS	<i>Serial Copy Management System</i>
SDTV	<i>Standard Definition Television</i>
SET	Sociedade Brasileira de Engenharia de Televisão
SGBD	Sistema Gerenciador de Banco de Dados
SPDIF	<i>Sony-Philips Digital Interface Format</i>
SQVGA	<i>Square Quarter Video Graphics Array</i>
STD	<i>Standard</i>
T-COMMERCE	<i>Television Electronic Commerce</i>
T-GOV	<i>Television Electronic Government</i>
T-HEALTH	Television Health
T-LEARNING	<i>Television Learning</i>
TV	Televisão
TXT	<i>Text /Texto</i>
UFPB	Universidade Federal da Paraíba
UHF	<i>Ultra High Frequency</i>
UIT	<i>International Telecommunication Union</i>
USB	<i>Universal Serial Bus</i>
VGA	<i>Video Graphics Array</i>
VHF	<i>Very High Frequency</i>
WiFi	<i>Wireless Fidelity</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
XHTML	<i>eXtensible Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	20
1.1 OBJETIVO GERAL.....	22
1.2 OBJETIVOS ESPECÍFICOS	23
1.3 JUSTIFICATIVA	23
1.4 ESTRUTURA DO TRABALHO	25
2 ASPECTOS FUNDAMENTAIS DA TV DIGITAL INTERATIVA.....	26
2.1 MODELO DE CONSUMO DE SERVIÇOS	31
2.2 ARQUITETURA DE CAMADAS GENÉRICA	34
2.3 O LADO DO TRANSMISSOR: PROCESSO DE DIFUSÃO E SUAS ETAPAS	36
2.4 O MECANISMO DE DIFUSÃO DE DADOS (DATACASTING).....	42
2.5 O LADO DO RECEPTOR: SET-TOP BOX E CAMADA DE MIDDLEWARE	45
2.6 SISTEMAS PIONEIROS DE TV DIGITAL	55
3 A INTERATIVIDADE NA TV DIGITAL	60
3.1 NÍVEIS DE INTERATIVIDADE NA TELEVISÃO	62
3.2 CLASSIFICAÇÃO DOS CONTEÚDOS E APLICAÇÕES INTERATIVAS	66
3.3 O CANAL DE RETORNO OU DE INTERATIVIDADE	68
4 SISTEMA BRASILEIRO DE TV DIGITAL E PADRÃO ISDB@TB	74
4.1 ABORDAGEM HISTÓRICA	74
4.2 CARACTERÍSTICAS TÉCNICAS	77
5 O MIDDLEWARE GINGA.....	85
5.1 SUBSISTEMA GINGA-CC.....	87
5.2 SUBSISTEMA GINGA-NCL	89
5.3 O SUBSISTEMA GINGA-J	91

5.4	MODELO DE APLICAÇÃO E AMBIENTE DE EXECUÇÃO GINGA-J.....	102
5.5	APLICAÇÕES XLET	105
5.6	A IMPLEMENTAÇÃO DE REFERÊNCIA OPENGINA	107
6	TRABALHOS CORRELATOS	109
6.1	PROTÓTIPO DE JOGO EDUCACIONAL QUIZ EDTEC	109
6.2	PROTÓTIPO DIGA DOCTOR.....	110
6.3	PROTÓTIPO T-COMMERCE BASEADO EM MHP E JAVA TV (DARÓS, 2004)...	111
6.4	PROTÓTIPO CONTROLE DE REDE DOMÉSTICA GINGA HOME.....	111
6.5	PROTÓTIPO BASEADO NO SUBSISTEMA GINGA-NCL (DUARTE, 2009)	112
7	PROTÓTIPO DE APLICAÇÃO T-COMMERCE BASEADO EM GINGA-J	114
7.1	METODOLOGIA UTILIZADA E MODELAGEM DO PROTÓTIPO.....	115
7.1.1.	Funcionamento do sistema e identificação de requisitos.....	118
7.1.2.	Análise e levantamento de requisitos	120
7.1.3.	Casos de uso levantados	122
7.1.4.	Diagramas de casos de uso	122
7.1.5.	Matriz de requisitos funcionais x casos de uso.....	124
7.2	DESENVOLVIMENTO DO PROTÓTIPO.....	124
7.2.1.	Servidor web.....	125
7.2.2.	Aplicação Xlet TCommerce	127
7.3	RESULTADOS OBTIDOS.....	141
	CONCLUSÃO.....	146
	REFERÊNCIAS	148
	APÊNDICE A – CASOS DE USO NO MODO EXPANDIDO	161
	APÊNDICE B – DIAGRAMA E-R DO BANCO DE DADOS DO SERVIDOR.....	172
	APÊNDICE C – INSTALANDO E CONFIGURANDO O AMBIENTE.....	173

APÊNDICE D – INSTRUÇÕES DE USO DO PROTÓTIPO.....	180
ANEXO A – LISTA DE PACOTES MÍNIMOS DO GINGA-J (ABNT-NBR 15606-4)	188
ANEXO B - PACOTES DTV UI (LWUIT) – STATUS DE IMPLEMENTAÇÃO	191

1 INTRODUÇÃO

A televisão digital aberta, embora não seja novidade em nações mais desenvolvidas, vem ganhando projeção aqui no Brasil na última década, com a implantação pelo Governo Federal do Comitê do Sistema Brasileiro de Televisão Digital (SBTVD), visando, a princípio, o desenvolvimento de um padrão de transmissão genuinamente nacional.

Quando da instituição do comitê, o Governo Federal estipulou algumas metas, como a adoção de um padrão de transmissão terrestre que possibilitasse fácil acesso por parte das camadas mais pobres, possibilidade de recepção por dispositivos os mais variados possíveis, bem como oferecesse mecanismos de interatividade.

O governo enxerga a televisão aberta como meio de ampla disseminação e instrumento em potencial de inclusão social e digital da grande parcela da população que não dispõe de recursos para usufruir outros. Considerando a ampla disseminação e aceitação da televisão aberta junto pela população brasileira em geral, principalmente nas camadas mais pobres, o potencial vislumbrado pelo governo é viável, desde que se desenvolvam estruturas que o suportem.

Além de oferecer conteúdo de imagem e som em alta definição, com qualidade bem superior àquela do sistema analógico, a televisão digital oferece a possibilidade de utilização de recursos inovadores, tais como multiprogramação e interatividade e transmissão de dados – mesmo que de forma unidirecional – via canal de difusão.

A interatividade, nesse contexto, representa uma quebra de paradigma, por possibilitar que o telespectador torne-se agente ativo e passe a interagir em um sistema onde antes não passava de mero observador. A possibilidade de interação transforma a televisão em elemento de interesse tanto governamental – como instrumento de inclusão digital e disseminador de conteúdo educacional e governamental em larga escala – como comercial,

tendo em vista a abrangência deste meio no contexto brasileiro.

Embora a proposta original do comitê fosse estabelecer um padrão exclusivamente nacional, em 2006, devido a diversas circunstâncias, definiu-se como padrão nacional o sistema japonês *Integrated Services Digital Broadcasting* (ISDB) com algumas melhorias, nascendo assim o padrão *Brazilian Integrated Services Digital Broadcasting Terrestrial* (ISDB@Tb).

Definido o padrão, buscou-se então a definição do *middleware* – camada de software que opera entre o sistema operacional do receptor e as aplicações - a ser adotado como referência e padrão no SBTVD. Foi adotado então o projeto Ginga, uma especificação aberta e fruto do trabalho de duas equipes: o laboratório TELEMÍDIA da Pontifícia Universidade Católica (PUC) do Rio de Janeiro e o Laboratório de Aplicações de Vídeo Digital (LAViD) da Universidade Federal da Paraíba (UFPB).

A estrutura do *middleware* Ginga normalmente é representada por um modelo dividido em três distintas partes: um subsistema de suporte ao desenvolvimento e processamento de aplicações declarativas (Ginga-NCL), um subsistema de suporte ao desenvolvimento e execução de aplicações procedurais desenvolvidas em plataforma Java (Ginga-J) e um núcleo comum (Ginga *Common Core*) que provê funcionalidades de suporte aos dois ambientes de aplicação além de promover um nível de abstração entre o hardware e o sistema operacional do receptor, executando funções de baixo nível.

Questões envolvendo discussões acerca de pagamento de *royalties* devido ao uso de bibliotecas proprietárias resultaram em atrasos no desenvolvimento do ambiente procedural do Ginga(Ginga-J), a cargo da equipe do LAViD. Entretanto, em março de 2008 o fórum SBTVD e a Sun Microsystems assinaram um memorando onde esta se comprometeu a desenvolver uma especificação aberta para substituir as bibliotecas proprietárias até então utilizadas.

Assim, no final de 2008 a Sun entregou a especificação JavaDTV para o fórum sem cobrança de *royalties*, substituindo as funcionalidades das bibliotecas proprietárias herdadas do GEM, favorecendo assim a utilização da plataforma Java e subsistema Ginga-J no processo de desenvolvimento de aplicações Ginga.

Em virtude das características já conhecidas da plataforma Java na tarefa de desenvolvimento de aplicações, optou-se neste trabalho pela utilização do subsistema procedural do Ginga quando do desenvolvimento do protótipo proposto como objetivo. No contexto do *middleware* Ginga, o modelo de arquitetura Java mostra-se viável para suportar o desenvolvimento de aplicações complexas, sejam clientes ou serviços, de modo particular envolvendo interatividade e distribuição em rede. Por ser uma área tecnológica relativamente recente, pouco explorada no Brasil e com potencial futuro, o desenvolvimento de aplicações Java para a TV Digital constitui uma área interessante a ser explorada pelos desenvolvedores de software.

Nesse ínterim, esta pesquisa tem como propostas: apresentar um estudo do ambiente de televisão digital, sua estrutura, conceitos e componentes envolvidos; estudar o Sistema Brasileiro de Televisão Digital – SBTVD e seu *middleware*, o Ginga; e por fim o desenvolvimento de um protótipo baseado no subsistema Ginga-J, explorando a interatividade oferecida pelo canal de retorno, utilizando um ambiente que emula o funcionamento de um receptor de televisão digital ou *set-top box*.

1.1 OBJETIVO GERAL

Desenvolver um protótipo de aplicação interativa com base na estrutura do ambiente de execução procedural do *middleware* Ginga – o subsistema Ginga-J, depois de realizado o estudo sobre a televisão digital interativa.

1.2 OBJETIVOS ESPECÍFICOS

Este projeto tem como objetivos específicos:

- a) estudar as características de cada módulo de um ambiente de televisão digital e a organização da estrutura como um todo;
- b) breve estudo dos padrões pioneiros de televisão digital: americano, europeu e japonês;
- c) compreender o funcionamento do processo de interatividade na televisão digital e os mecanismos necessários envolvidos;
- d) estudar o padrão nipo-brasileiro ISDB-Tb e seu *middleware* Ginga;
- e) desenvolver um protótipo tendo como base a implementação de referência do ambiente procedural do *middleware* brasileiro, explorando o funcionamento do canal de retorno como mecanismo de interatividade.

1.3 JUSTIFICATIVA

A presença de um mecanismo que ofereça interatividade ao usuário potencializa o interesse pelo sistema televisivo, de modo geral. Para o usuário, o advento da televisão digital interativa - fruto da chamada “convergência tecnológica” envolvendo a televisão, sistemas computacionais e redes de telecomunicações – representa o surgimento de uma nova mídia, com maior variedade de conteúdos e ofertando além de serviços, um mecanismo de interação que o transforma em participante do sistema (BREUNIG; MARQUES, 2008; PASCHOAL NETO; CARVALHO, 2008).

O governo por sua vez pretende utilizar o meio como agente de inclusão social e digital, devido à ampla cobertura da televisão aberta no país. Já o setor privado, de modo

especial no que se refere ao comércio, poderia dispor de um novo canal de comunicação direta com o cliente, com propaganda em larga escala e até mesmo comércio eletrônico: desde que haja um canal de retorno e uma estrutura voltada a esta funcionalidade, o usuário poderá comprar *online* e na tela da TV, enquanto assiste à novela das oito, e uma chamada de *merchandising*¹ aparece no canto da tela, entre outras possibilidades. Esta nova modalidade de comércio vem sendo chamada de *Television Electronic Commerce (T-Commerce)* (BREUNIG; MARQUES, 2008; PASCHOAL NETO; CARVALHO, 2008).

A TV Digital, do ponto de vista tecnológico, é ainda pouco explorada no Brasil – resumindo-se nas frentes de pesquisa para desenvolvimento do padrão de TV e *middleware* – e com grande potencial de exploração futura. A exploração deste ambiente significa oportunidades de negócios para empresas de software e conseqüentemente, um campo promissor de trabalho para desenvolvedores, especialmente no que se refere à aplicações cliente que rodem no ambiente do *middleware* Ginga, em virtude da extensão da comunidade Java no Brasil. Isto sugere que uma vez a interatividade na TV Digital se consolide, a demanda por aplicações televisivas interativas – seja de que tipo for - será alta, para que possa atender às necessidades governamentais e comerciais

No contexto da TV Digital, o ambiente de desenvolvimento procedural do *middleware* Ginga, baseada em Java, mostra-se muito interessante como campo de trabalho. As características naturais da plataforma Java – entre as quais podemos citar a alta granularidade, portabilidade de código, ênfase em processamento distribuído, entre outras – dão suporte a várias funcionalidades exigidas pela especificação Ginga, como acesso a recursos de baixo nível da plataforma, controle do ciclo de vida da aplicação, suporte à transação segura de dados usando o canal de interatividade, persistência de conteúdo, entre outros (BOQUIMPANI, 2009; LAGALLY; PÄTZOLD, 2009; NERY e SILVA et al, 2007).

¹ Do termo francês “*merchand*”, relaciona-se a operações com mercadorias (MERCHANDISING, 2010).

Por sua vez, as linguagens procedurais – sendo Java uma delas - devido à sua natureza, permitem um maior controle do código por parte do programador. Ao mesmo tempo em que este pode lançar mão de uma série de recursos, o grau de dificuldade neste tipo de paradigma de programação é maior do que nas linguagens declarativas (NERY e SILVA et al, 2007).

Considerando-se as características da plataforma Java, o ambiente proporcionado pela parte procedural do Ginga – o subsistema Ginga-J que atende à recomendação ITU J.202 oferecendo compatibilidade com outros *middlewares* já existentes –, bem como por se tratar de uma nova tecnologia, com grande potencial de exploração comercial, aliados ao fato de termos em nosso país uma comunidade consideravelmente grande e crescente de desenvolvedores Java, o desenvolvimento de um protótipo baseado no ambiente procedural, após a conclusão dos estudos sobre a TV Digital e o *middleware* Ginga, mostra-se viável.

1.4 ESTRUTURA DO TRABALHO

O trabalho envolve um estudo prévio sobre a televisão digital e foi estruturado da seguinte forma: no Capítulo 2 serão abordadas as características técnicas, mecanismos componentes e funcionamento do ambiente de televisão digital, bem como de forma breve os sistemas pioneiros. No Capítulo 3 trata a questão da interatividade na televisão digital. O Capítulo 4 apresenta o Sistema Brasileiro de Televisão Digital, trazendo um levantamento histórico e suas principais características. O *middleware* Ginga e seus subsistemas serão abordados no Capítulo 5, bem como o funcionamento das aplicações Java para TV Digital (Xlets), também apresentando brevemente a implementação de referência OpenGinga. Os trabalhos correlatos são tratados no Capítulo 6. O Capítulo 7 é o ápice do trabalho, apresentando as etapas de projeto, desenvolvimento e testes com o protótipo proposto.

2 ASPECTOS FUNDAMENTAIS DA TV DIGITAL INTERATIVA

A televisão, desde o seu surgimento em 1936 por ocasião da implantação do canal *British Broadcast Corporation* (BBC) de Londres, consolidou-se como um dos mais abrangentes meios de comunicação, disponibilizando aos telespectadores informações e entretenimento, inserindo-se completamente no ambiente e rotina familiar (BRACKMANN; SILVA; LUZZARDI, 2008; DARÓS, 2004; PASCHOAL NETO; CARVALHO, 2008). No Brasil, a televisão foi implantada por iniciativa do jornalista Francisco de Assis Chateaubriand Bandeira de Melo, que mandou buscar nos Estados Unidos a aparelhagem necessária, bem como 200 aparelhos receptores, inaugurando no dia 18 de setembro de 1950 a TV Difusora Canal 3 - mais tarde renomeada como TV Tupi - em São Paulo, passando a ser o quarto país do mundo a implantar a televisão (CARVALHO, 2008; OLIVEIRA, 2008).

No percurso do tempo, a televisão passou por transformações tecnológicas significativas, de modo a responder os anseios de uma sociedade cada vez mais exigente (BRACKMANN; SILVA; LUZZARDI, 2008; CRUZ, 2009; MONTEZ; BECKER, 2005). As primeiras grandes transformações foram: o surgimento da TV em cores nos anos 50, nos Estados Unidos e na Europa, seguido pela criação do controle remoto, devido ao aumento considerável no número de canais, caracterizado como o primeiro componente digital integrado ao aparelho de televisão (DUARTE, 2009; KNEIPP, 2009).

Mais adiante, iniciou-se o processo de digitalização no ambiente de produção da TV, com a substituição gradual da tecnologia analógica por sistemas computadorizados, equipamentos e estúdios digitais, facilitando o trabalho de criação e edição nos estúdios em virtude da disposição de novos recursos tecnológicos, flexibilidade e automatização das tarefas (ARAÚJO, 2009; MONTEZ; BECKER, 2005).

Nos anos 70, pesquisadores japoneses do *Nippon Hōsō Kyōkai* (NHK) *Science &*

Technical Research Laboratories deram início ao processo de desenvolvimento da TV de alta definição, oferecendo uma recepção com o dobro de linhas em relação à TV convencional, passando a ser conhecida como padrão *High Definition Television* (HDTV) (CARVALHO, 2008; CRUZ, 2009).

A partir dos anos 80, iniciou-se um processo que desencadeou a inovação mais significativa na cadeia televisiva: a substituição da difusão em sinal analógico pela transmissão em sinal digital, embora a produção de conteúdos já viesse sendo feita em formato digital há bastante tempo, assim como a recepção em alta definição (AMATO NETO, 2006; DARÓS, 2004; LIMA JUNIOR; GARRIDO, 2009). Os japoneses iniciaram as pesquisas em torno da digitalização do meio de transmissão, seguidos pelos americanos e europeus. A implantação do primeiro sistema de transmissão em modo digital deu-se nos Estados Unidos na década de 90, vindo em seguida a Europa, e em 2003, o Japão, sendo que atualmente o processo de digitalização nos demais países do mundo vem crescendo de forma significativa (BRACKMANN; SILVA; LUZZARDI, 2008; DUARTE, 2009; MONTEZ; BECKER, 2005).

A transmissão do sinal, seja via satélite, cabo ou difusão terrestre, passa assim a ser feita em formato de código binário – sequências de dígitos 0 e 1 – permitindo a transmissão de conteúdo não somente audiovisual, mas também de fluxos de dados. Logo, possibilita a recepção por dispositivos eletrônicos com capacidade computacional, permitindo assim a oferta de uma gama de novos serviços ao telespectador, graças à difusão de dados computáveis (aplicações) junto aos fluxos de áudio e vídeo (BREUNIG; MARQUES, 2008; LIMA JUNIOR; GARRIDO, 2009; MONTEZ; BECKER, 2005).

Em um primeiro instante, a alteração perceptível do ponto de vista do telespectador refere-se à melhoria na qualidade do áudio e vídeo transmitidos. Enquanto a TV analógica disponibiliza no máximo dois canais de áudio estéreo, os padrões digitais oferecem

som com qualidade de *Compact Disk* (CD), com até cinco canais de áudio e um canal de *woofer*², proporcionando um efeito mais realístico - efeito *surround*³ (ANDREATA, 2006; BREUNIG; MARQUES, 2008; CRUZ, 2009; DUARTE, 2009).

Devido à codificação digital, o sinal apresenta maior imunidade a interferências, reduzindo a ocorrência de problemas como chuviscos e “efeito fantasma“, além da possibilidade de se aplicar técnicas digitais de correção e recuperação do sinal. (BECKER; MONTEZ, 2004; MONTEZ; BECKER, 2005; PASCHOAL NETO; CARVALHO, 2008). Além do aspecto de imunidade, o padrão digital oferece um incremento na qualidade de resolução da imagem em relação à TV analógica convencional, tanto em quantidade de linhas e pixels disponível como na relação de aspecto ou proporcionalidade (*aspect ratio*), oferecendo ao telespectador quatro níveis de resolução (BREUNIG; MARQUES, 2008):

- a) HDTV: alta definição, com formato de tela 16:9 (*widescreen*), resolução de 1080 linhas x 1920 *pixels* por linha em varredura progressiva ou entrelaçada, ou 720 linhas x 1280 *pixels* por linha em varredura progressiva;
- b) *Enhanced Definition Television* (EDTV): definição estendida, com formato de tela 16:9 e resolução de 480 linhas x 720 *pixels* por linha em varredura progressiva;
- c) SDTV: definição padrão, com formatos de tela 4:3 (similar ao modelo analógico) ou 16:9, com resolução de 480 linhas x 640 *pixels* por linha com varredura entrelaçada;
- d) *Low Definition Television* (LDTV): baixa definição, com formato de tela 4:3 e resolução de 240 linhas x 320 *pixels* por linha em varredura progressiva.

Ainda segundo Breunig e Marques (2008) são denominados *Full-HD* os aparelhos receptores capazes de operar na resolução máxima de HDTV – 1080 linhas com 1920 *pixels*

² Alto-falante utilizado para reproduzir frequências de som graves e médias (WOOFER, 2010).

³ Conceito de expansão da imagem do som a três dimensões, proporcionando maior realismo (SOM SURROUND, 2010).

por linha, com varredura progressiva, sendo esta modalidade de varredura considerada superior à varredura entrelaçada e mais indicada aos televisores com telas grandes.

Além dos aspectos relacionados à qualidade de áudio e vídeo, a TV Digital oferece melhorias significativas em outros aspectos, como possibilitar a interoperabilidade com outras tecnologias – pelo fato de ser codificada digitalmente – e oferecer a possibilidade de interatividade para o telespectador (CARVALHO, 2008). Resumidamente, podemos listar alguns dos benefícios (BREUNIG; MARQUES, 2008; CARVALHO, 2008; KNEIPP, 2009; MONTEZ; BECKER, 2005):

- a) possibilidade de compactação do sinal, a despeito do modo analógico, permitindo um melhor aproveitamento da banda de transmissão (normalmente na faixa de largura de 6 MHz) em virtude da aplicação de técnicas digitais visando diminuir a taxa de bits da transmissão. Esse melhor aproveitamento do espectro possibilita a transmissão de conteúdo simultâneo - até quatro canais em resolução *Standard Definition Television* (SDTV) - onde antes era veiculado apenas um canal, e é chamado de *multiprogramação*, ilustrado na Figura 1:

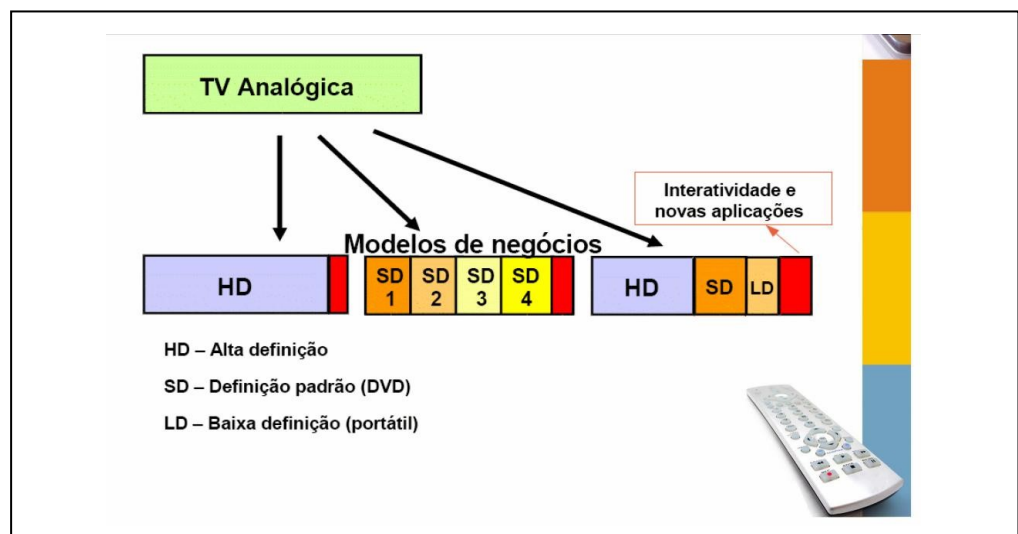


Figura 1. Divisão do espectro de transmissão, possibilitando a multiprogramação
Fonte: ARAUJO, R. (2009, p. 20)

- b) sinal mais robusto, possibilitando uma transmissão mais imune a ruídos, fantasmas e chuviscos, graças às técnicas de modulação digital que permitem a utilização de algoritmos de detecção e correção de erros na transmissão;
- c) ausência de interferência entre os canais, eliminando o emprego de canais livres, técnica utilizada na transmissão analógica, onde entre dois canais de transmissão é deixada uma faixa de espectro livre, para evitar ou ao menos diminuir a interferência entre as transmissões. Isto contribui para a otimização do espectro e possibilita maior oferta de canais;
- d) possibilidade de envio de múltiplos fluxos de áudio em um mesmo programa;
- e) *datacasting*: possibilidade de envio de informações como textos, legendas e arquivos contendo dados, em meio ao conteúdo audiovisual;
- f) recursos de *interatividade*: possibilidade de envio de conteúdos aplicativos que permitem a interação do usuário, como guia de programação, aplicações bancárias, comércio eletrônico, entre outros;
- g) integração com outras mídias e meios de comunicação digital, como a Internet;
- h) acessibilidade: possibilidade de armazenar localmente o conteúdo transmitido, bem como acessar conteúdos sob demanda;
- i) possibilidade de interação entre o aparelho receptor e outros dispositivos, como o computador, por exemplo, permitindo a composição de uma rede doméstica;
- j) portabilidade e mobilidade: possibilidade de recepção do sinal por dispositivos móveis como celulares, laptops, *Personal Digital Assistant* (PDA's) e outros dispositivos de tamanho limitado.

Dentre as vantagens oferecidas, a interatividade – interação ativa entre o usuário e a programação – é sem dúvida o aspecto mais revolucionário e impactante possibilitado pela TV Digital (PASCHOAL NETO; CARVALHO, 2008). Graças ao mecanismo de

interatividade, desde que haja um canal de retorno disponível entre o receptor e o transmissor, o telespectador deixa de ser o elemento passivo característico do modelo convencional (onde a interação se resume a alternar entre canais via controle remoto) passando a ser agente ativo e participativo. Poderá exercer poder de escolha do conteúdo programático transmitido, acesso a uma variedade de recursos inovadores – comércio eletrônico, conteúdo digital e Internet, por exemplo – além da possibilidade de influir de maneira mais acentuada no processo de geração de conteúdo e programação (KNEIPP, 2009; MONTEZ; BECKER, 2005; OLIVEIRA, 2008).

A interatividade engloba aspectos tecnológicos e sociais complexos, e será abordada de modo mais profundo em uma seção específica deste trabalho. A Tabela 1 traça um comparativo entre a TV analógica e a TV Digital:

Tabela 1. Principais diferenças entre TV analógica e TV Digital

FATOR	ANALÓGICO	DIGITAL
Qualidade da imagem	Boa	Excelente (DVD, cinema)
	Degradações: fantasma e chuvisco	Não degrada enquanto o sinal puder ser recebido
Qualidade do som	Mono ou estéreo	Estéreo ou <i>surround</i>
Novos recursos		Novos serviços (cultura digital)
		Múltiplos canais de áudio e vídeo
Otimização do espectro	Uso do espectro limitado por interferências	Possível uso de canais adjacentes

Fonte: PASCHOAL NETO, J.; CARVALHO, J. (2008, p. 62)

2.1 MODELO DE CONSUMO DE SERVIÇOS

O ambiente de TV Digital interativa, analisada sob o ponto de vista de um sistema envolvendo a participação de agentes, costuma ser dividido em três partes integrantes, listadas

a seguir e representados na Figura 2 (MONTEZ; BECKER, 2005; SCHROEDER, 2007):

- a) difusor ou provedor de conteúdo: responsável pela geração do conteúdo, compreendendo som, imagem e aplicações, além de oferecer suporte à possível interação vinda por parte do telespectador/usuário;
- b) transmissor/meio de transmissão: o transmissor recebe o conteúdo vindo do sistema difusor, realiza o processo de multiplexação para geração de um fluxo de transporte e envia aos usuários pelo meio de transmissão (canal de difusão via *broadcast*⁴). As interações enviadas pelo usuário são possíveis desde que se empregue um meio de transmissão denominado canal de retorno;
- c) receptor: recebe e apresenta o conteúdo (imagem, som e aplicações ou serviços) ao telespectador, além de oferecer mecanismos de interação/interatividade para o usuário, graças ao suporte oferecido pelo *set-top box*, que será visto adiante.

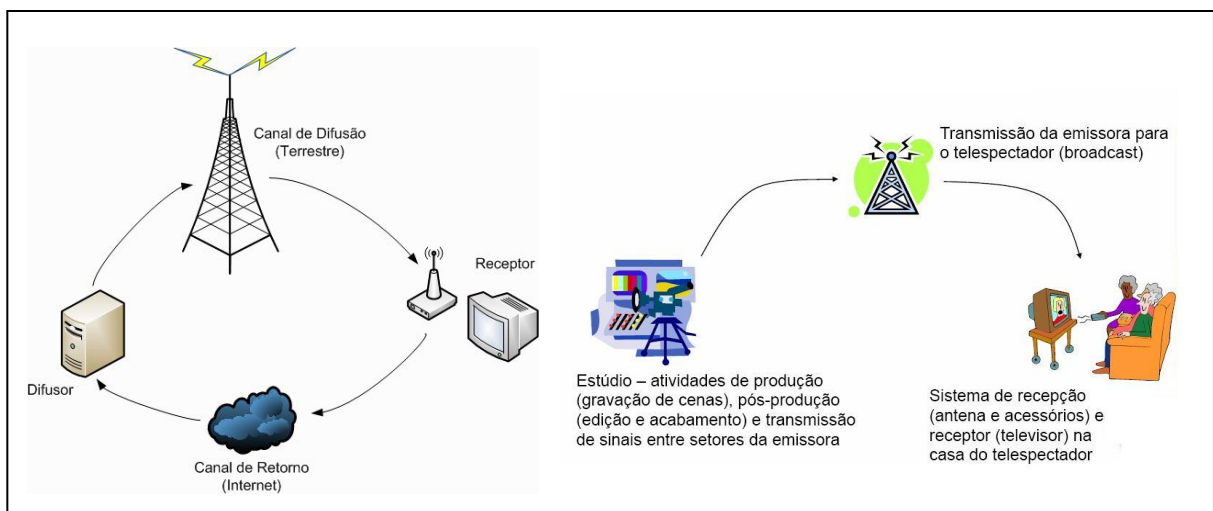


Figura 2. Visualização do sistema de TV Digital e integração entre as partes envolvidas
 Fonte: Adaptado de MONTEIRO, B. (2009, p. 23) e ARAUJO, R. (2009, p. 08)

O processo produtivo da TV Digital é caracterizado pelo surgimento de novos conceitos relacionados à produção de mídia nos estúdios, formando um modelo de abstração

⁴ Processo de difusão via ondas de rádio, enviada a vários receptores simultaneamente (BROADCAST, 2010).

composto por eventos, programas, serviços e o conjunto *bouquet*, que será visto adiante.

Neste modelo abstrato, um evento é o elemento base ou atômico de produção de mídia, constituído pelo agrupamento de *streams* de áudio, vídeo e dados com início e fim estipulados – um bloco de um programa televisivo, por exemplo. A concatenação de vários eventos forma um programa, como um capítulo de novela. Por sua vez, um serviço é considerado a principal unidade de produção e consumo da TV Digital, constituído por uma sequência de programas, controlada pelo estúdio difusor (DARÓS, 2004; PAES; ANTONIAZZI, 2005).

Por fim, tem-se o *bouquet*, no qual serviços produzidos por estúdios diversos são arranjados em um conjunto, organizado e distribuído por uma central de produções via um ou mais meios de transmissão (DARÓS, 2004; PAES; ANTONIAZZI, 2005). Na Figura 3 podemos observar o agrupamento e compartilhamento de serviços em *bouquet* entre diferentes meios de transmissão:

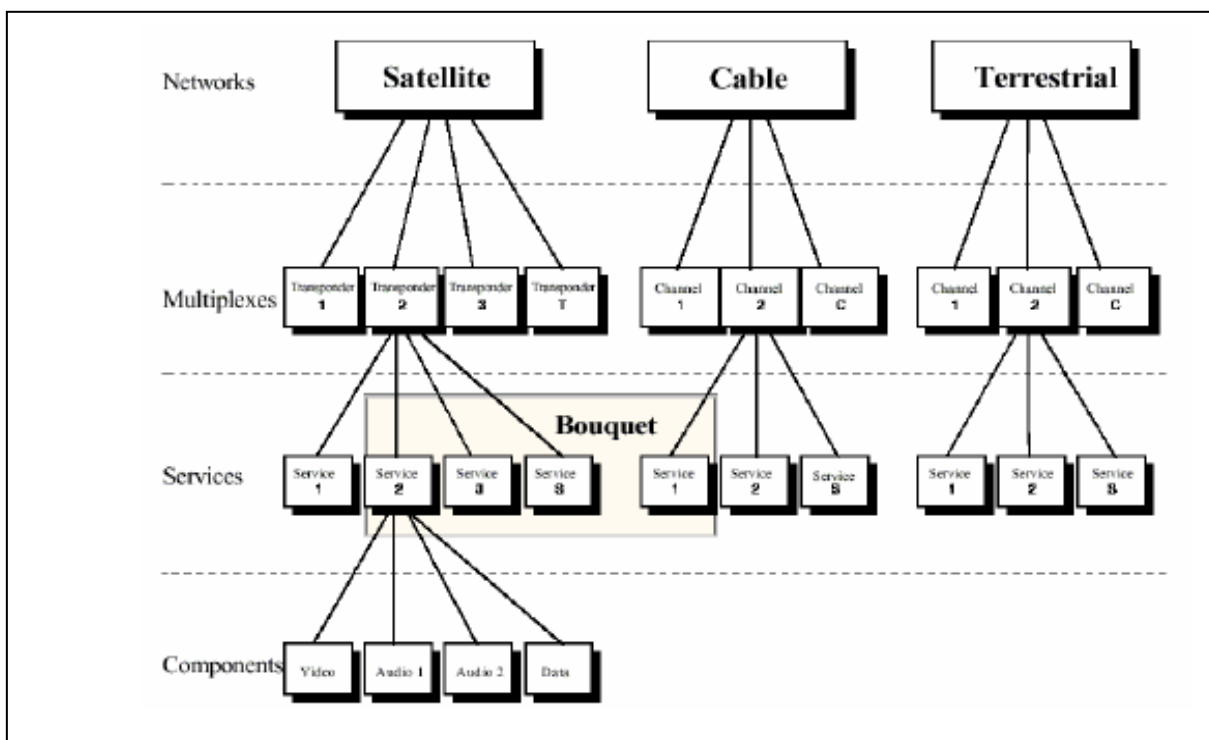


Figura 3. Organização em *bouquet*, compartilhado entre diferentes meios de transmissão

Fonte: DARÓS, J. (2004, p. 15) sddsdffsdff--.aaa-aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaadff

Para suprir as novas necessidades exigidas no processo de produção de mídia da TV Digital interativa, um novo conceito de estúdio especializado passou a fazer parte do processo produtivo das emissoras: o estúdio de dados, tendo como funções a produção de dados (geração de videotexto e conteúdos HTML) e a produção de aplicações, envolvendo o desenvolvimento de *software*. Estes conteúdos serão inseridos no fluxo de transmissão, junto aos sinais de áudio e vídeo, durante o processo de multiplexação, que será abordado adiante (DARÓS, 2004).

2.2 ARQUITETURA DE CAMADAS GENÉRICA

O conteúdo disponibilizado pelos estúdios precisa passar por uma série de etapas até ser entregue ao usuário em formato multimídia. De forma análoga ao modelo padronizado *Open Systems Interconnection* (OSI) das redes computacionais, o processo de manipulação dos conteúdos da TV Digital interativa, entre a produção no estúdio até ser exibido no aparelho receptor pode ser organizado em uma arquitetura de camadas autônomas dispostas hierarquicamente, presente nas duas extremidades do sistema - difusor e receptor (MACIEL; SILVA JUNIOR; SOUZA, 2009; PAES; ANTONIAZZI, 2005).

Nesse modelo, cada camada dispõe de serviços oferecidos por uma camada inferior a ela, executa as operações que lhe cabem sobre o conteúdo e remete o resultado à camada superior (MACIEL; SILVA JUNIOR; SOUZA, 2009; PAES; ANTONIAZZI, 2005).

A Figura 4 apresenta as cinco camadas que compõem a arquitetura genérica de um sistema de TV Digital interativa, segundo a visão análoga ao modelo OSI, bem como as diversas tecnologias ou padrões que compõe os serviços oferecidos por cada camada:

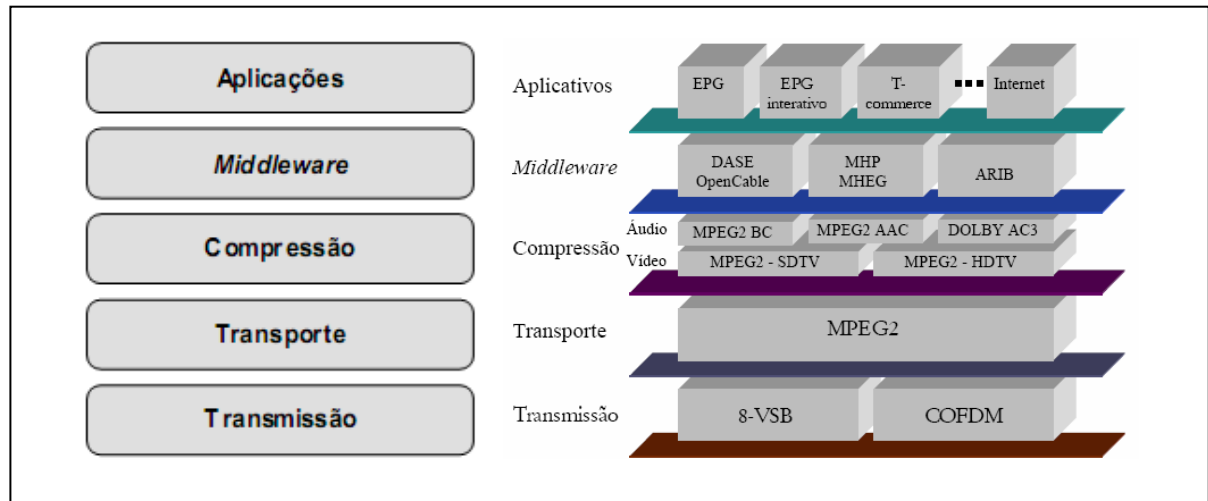


Figura 4. Arquitetura de camadas de um sistema de TV Digital e tecnologias envolvidas
 Fonte: Adaptado de MACIEL, D.; SILVA JUNIOR, L.; SOUZA, C. (2009, p. 02) e CAPDA (2004, p. 22).

Resumidamente, as funções das camadas dispostas na Figura 4 são (MACIEL; SILVA JUNIOR; SOUZA, 2009):

- a) transmissão: agrega três subsistemas sendo:
 - transmissão/recepção: levantamento do sinal no difusor e pela sintonia do sinal no receptor. O sinal é ampliado e enviado por um meio de transmissão, normalmente via cabo, satélite ou difusão terrestre;
 - modulação/demodulação: responsável pela modulação e demodulação do fluxo de transporte codificado;
 - codificação/decodificação – responsável pela codificação e decodificação (*codec*⁵) do fluxo de transporte.
- b) transporte: no lado emissor, é responsável pela multiplexação de vários programas (áudio, vídeo e dados) em um único fluxo de transporte. No lado receptor, realiza a demultiplexação do fluxo de transporte de acordo com o programa selecionado pelo telespectador;

⁵ Acrônimo de codificador/decodificador; componente ou dispositivo de software ou hardware que codifica/decodifica sinais (CODEC, 2010).

- c) compressão: realiza os processos de compressão de sinais de áudio e vídeo no difusor e descompressão no ambiente do telespectador;
- d) *middleware*: camada de software que oferece um serviço padronizado para a camada de aplicação, escondendo as peculiaridades e heterogeneidades das camadas de compressão, transporte e transmissão;
- e) aplicativos: visível para o usuário e que fará a interação direta com o mesmo, sendo suportada pelas camadas inferiores. É responsável pela execução dos aplicativos.

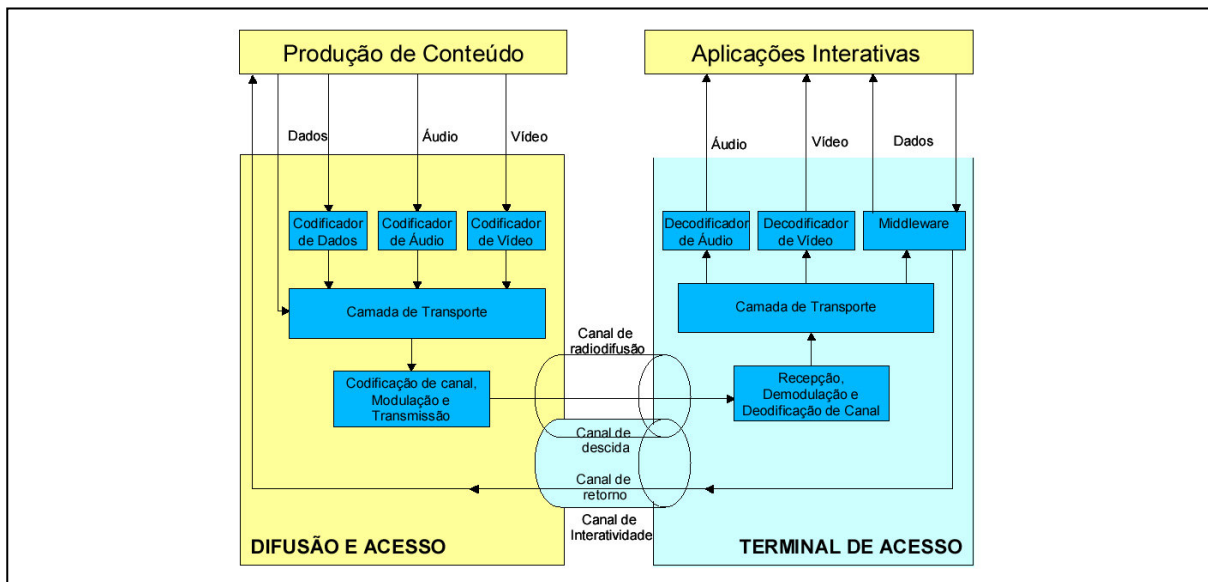


Figura 5. Diagrama do fluxo de informação entre as camadas do modelo genérico
 Fonte: ARNOLDO, M. (2008, p. 03)

2.3 O LADO DO TRANSMISSOR: PROCESSO DE DIFUSÃO E SUAS ETAPAS

Além do modelo genérico apresentado, a União Internacional de Telecomunicações (ITU) especificou um modelo contendo três blocos, voltado para a representação do processo de difusão, isto é, somente as etapas correspondentes às camadas de transmissão, transporte e compressão do modelo anterior. Este modelo de referência do

ITU é adotado pelos padrões de TV Digital em funcionamento atualmente, e pode ser observado na Figura 6 (ALVES, 2008; CARVALHO, 2008):

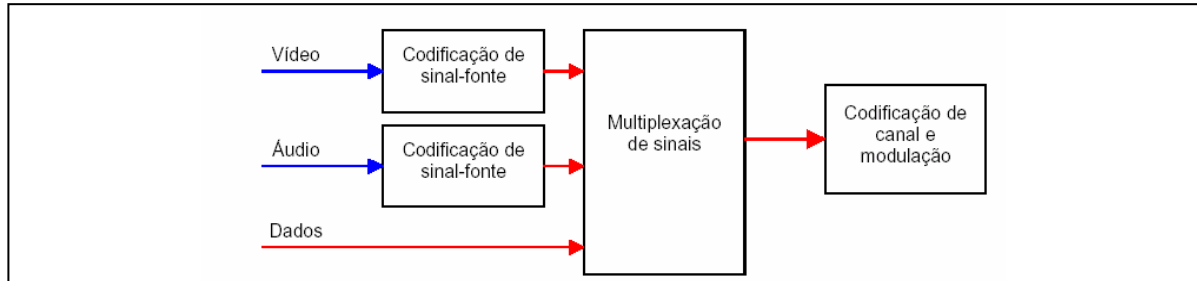


Figura 6. Modelo ITU-T para composição e transmissão do sinal de TV Digital
Fonte: CARVALHO, E. (2008, p. 17)

De acordo com o modelo apresentado, o processo de difusão subdivide-se em: a) codificação/compressão do sinal fonte; b) multiplexação dos serviços; e c) modulação do sinal digital. Antes de sofrer estes processos, os sinais são devidamente capturados na fonte analógica e digitalizados utilizando-se técnicas de quantização e amostragem, conforme ilustrado abaixo:

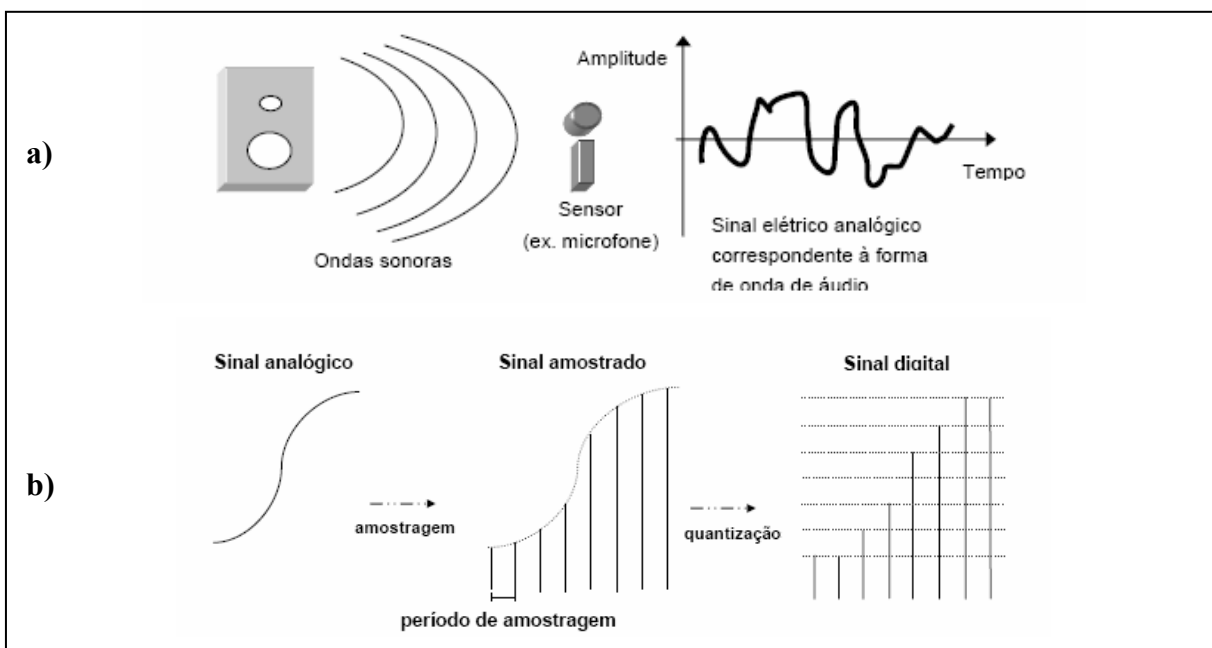


Figura 7. a) captura de sinais analógicos; b) amostragem e quantização do sinal
Fonte: Adaptado de BECKER, V.; MONTEZ, C. (2004, p. 13-14)

Na etapa de codificação e compressão, os sinais de áudio e vídeo analógicos, amostrados e quantizados durante o processo de captura, são transformados em conteúdos de formato digital graças à codificações padronizadas, dando origem aos fluxos elementares (*elementary stream*) de áudio e vídeo. Esta codificação precisa ser feita de maneira otimizada, de modo que os fluxos sejam gerados ocupando menos espaço possível preservando-se a qualidade, pois os recursos de largura de banda, processamento e capacidade de armazenamento tendem a ser limitados quando se trata de conteúdos multimídia – estes tendem a gerar volumes de dados enormes (ALVES, 2008; CARVALHO, 2008; MONTEIRO, 2009).

A compressão pode ser feita empregando-se abordagens algorítmicas de eliminação de redundâncias nos dados como também com base na percepção humana (mascaramento), quando aspectos não perceptíveis por humanos (como faixas de sons inaudíveis) são eliminados do espectro do sinal. Os padrões de codificação/decodificação (*codecs*) normalmente empregados na TV Digital são MPEG-2 e MPEG-3, da família *Moving Picture Experts Group* (MPEG), para áudio e/ou vídeo, e o padrão *Dolby Audio Compression 3* (AC3) para áudio (ALVES, 2008; CARVALHO, 2008; MONTEIRO, 2009).

A família de padrões MPEG é fruto do trabalho do consórcio *Moving Picture Experts Group* em busca da padronização dos processos de codificação, compressão e transporte de dados multimídia. Os protocolos definidos pela família MPEG são abertos e aceitos mundialmente em processos envolvendo componentes multimídia, sendo que a família é composta por vários protocolos que atendem a finalidades diversas (BECKER; MONTEZ, 2004; CARVALHO, 2008).

Na etapa de multiplexação, os fluxos elementares de áudio, vídeo e dados vindos da etapa anterior são reunidos em um único fluxo de saída intercalado pela da passagem dessas *streams* por um mecanismo multiplexador de base comum de tempo, de modo

sincronizado. Primeiramente, os fluxos elementares são agrupados logicamente, dando origem a um programa ou serviço. A seguir, um conjunto de serviços – programação – sofre a multiplexação, originando um feixe único encapsulado denominado fluxo de transporte - *transport stream*, composto por pacotes de 188 bytes de tamanho cada (BECKER; MONTEZ, 2004; MONTEIRO, 2009). A Figura 8 ilustra um exemplo de multiplexação de fluxos:

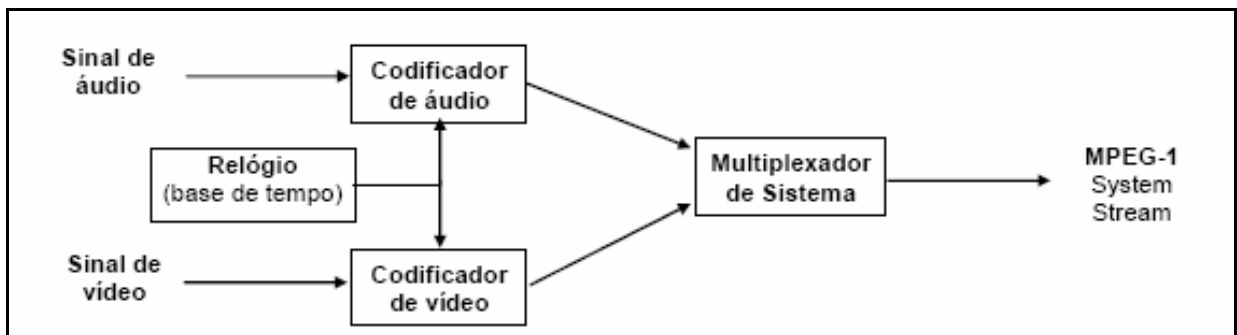


Figura 8. Multiplexação de áudio e vídeo em um *stream* MPEG-1 System
 Fonte: BECKER, V.; MONTEZ, C. (2004, p. 24)

Além de transportar o conteúdo “útil”, em cada pacote é reservado um espaço de 4 bytes de cabeçalho contendo metadados relativos ao conteúdo, como números de identificação de cada *stream* elementar - *Packetized Elementary Stream* (PES) - e serviço ou programa do qual pertence o pacote - *Process Identifier* (PID) -, formando um sistema de tabelas indexatórias - *Program Association Table* (PAT). Graças a estas estruturas as aplicações transmitidas e seus pacotes são catalogados, organizados e agrupados em conjuntos, de modo a orientar e facilitar o processo de demultiplexação/decodificação dos fluxos recebidos pelo aparelho receptor, reconstruindo o fluxo de informações de forma sincronizada e ordenada, antes de ser exibido. Esta organização facilita também a montagem de índices de conteúdo, orientando o processo de navegação entre serviços disponíveis e seleção de aplicações pelo usuário, no aparelho receptor (ABNT, 2010; BECKER et al, 2005). A Figura 9 exemplifica parte do processo de indexação dos pacotes:

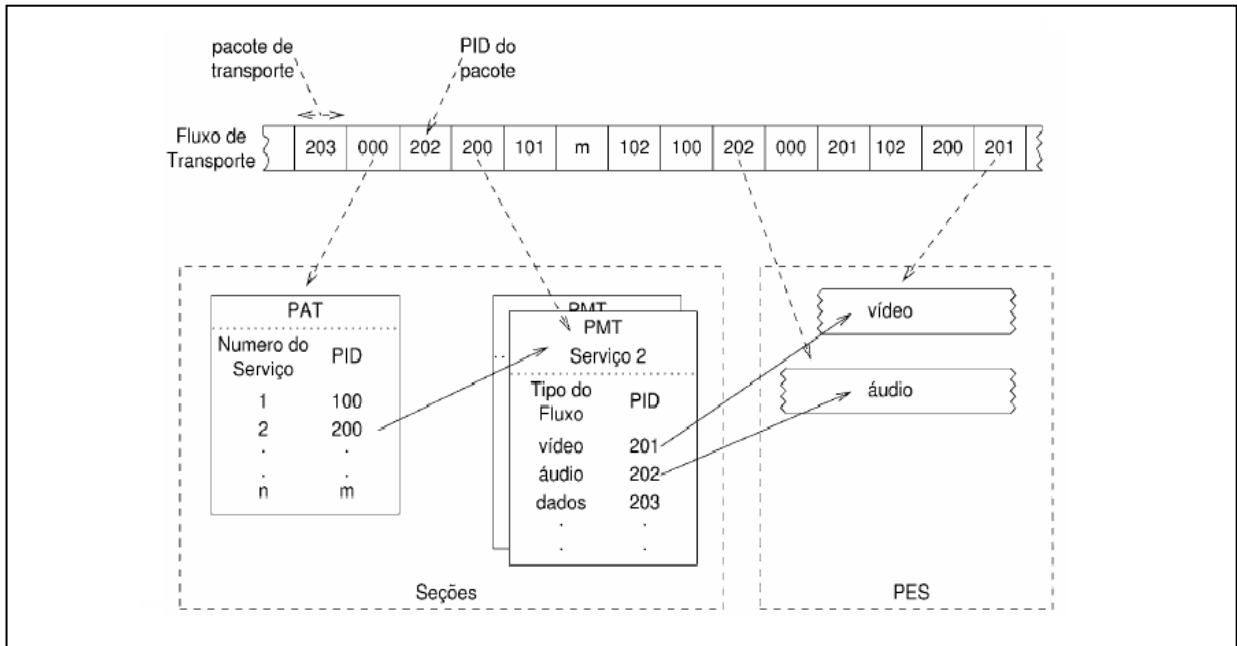


Figura 9. Fluxo de transporte contendo as tabelas PSI básicas e fluxos elementares
 Fonte: BECKER, V. et al (2005, p. 09)

O processo de multiplexação é realizado com base na especificação MPEG-2 *System: Transport Stream*, protocolo da família MPEG que define os serviços de fluxo de transporte (MONTEIRO, 2009). Na TV Digital interativa a etapa de multiplexação sofreu um acréscimo importante: agora não apenas áudio e vídeo são encapsulados em um fluxo, mas também dados ou aplicações, em um processo chamado de difusão de dados ou *datacasting*, graças ao mecanismo lógico denominado carrossel de dados, que será abordado de forma mais detalhada em breve. A Figura 10 ilustra o fluxo de transporte MPEG-2 *System-TS*:

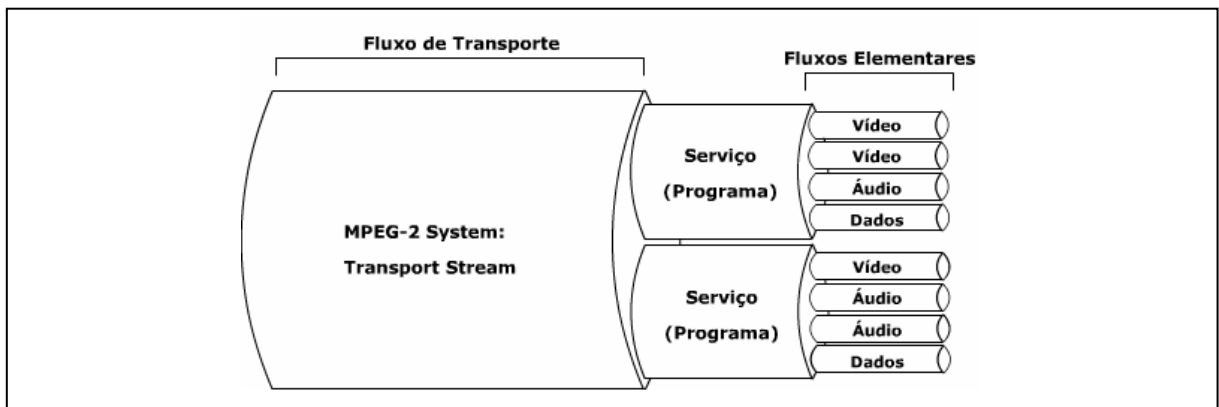


Figura 10. Fluxo de transporte e seus subfluxos
 Fonte: MONTEIRO, B. (2009, p. 12)

Antes de ser difundido pelos meios de transmissão, o fluxo de transporte é submetido ao processo de modulação. Nesta fase, a frequência original do sinal de dados do fluxo é deslocada, variando em amplitude, fase ou frequência, de acordo com um sinal auxiliar ou de referência denominado “portadora” – Figura 11. A portadora opera no canal de transmissão em uma faixa de frequência controlada, definida de forma a minimizar a influência de interferências e distorções presentes no sistema de transmissão (MONTEIRO, 2009; MONTEZ; BECKER, 2005; SILVA, 2003).

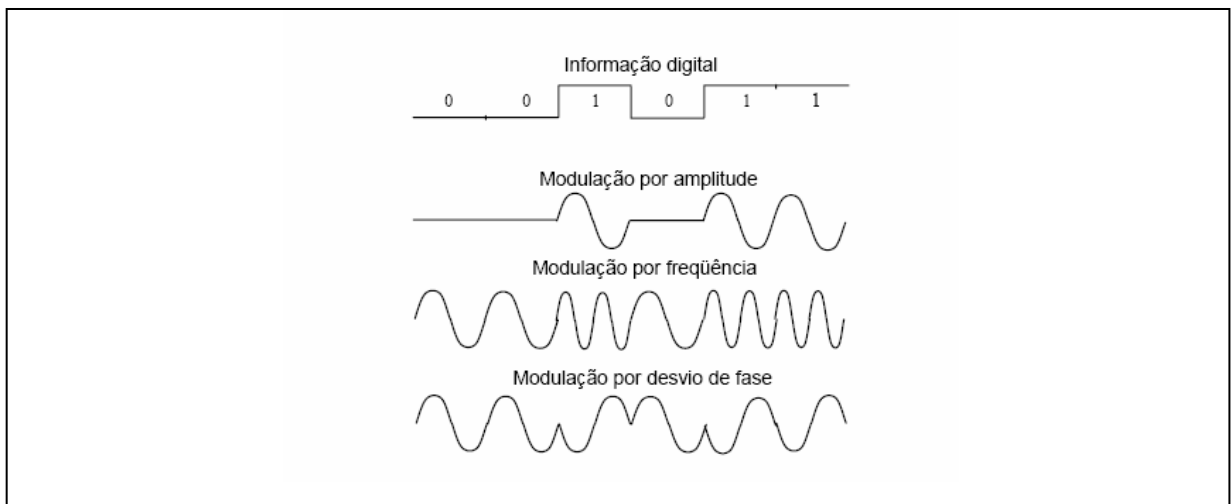


Figura 11. Modulação por amplitude, frequência ou fase
 Fonte: MONTEZ, V.; BECKER, C. (2005, p. 47)

Na TV Digital, os padrões de modulação mais empregados são o *8 Vestigial Side Band* (8-VSB) e *Coded Orthogonal Frequency Division* (COFDM) para modulação via difusão terrestre. Também são utilizados os padrões *Quadrature Phase Shift Keying* (QPSK) e *Quadrature Amplitude Modulation* (QAM) comumente utilizados em meios de transmissão via satélite e cabo, respectivamente (ANDREATA, 2006). O sinal, devido à baixa frequência em que é gerado pelo modulador, precisa ser amplificado antes de ser transmitido, de modo que seja forte o suficiente para se propagar pelo meio de transmissão. Para isso, utiliza-se um equipamento especial denominado *UpConverter* (BECKER; MONTEZ, 2004).

2.4 O MECANISMO DE DIFUSÃO DE DADOS (DATACASTING)

Uma das principais inovações que caracterizam o ambiente de TV Digital é a possibilidade de se difundir dados (informações, aplicativos e serviços) no mesmo sinal por onde são transmitidos áudio e vídeo, de forma concomitante, graças aos mecanismos de difusão de dados ou *datacasting*, constituindo a base para o processo de interatividade no ambiente de TV Digital (BECKER et al, 2005; BRACKMANN, 2008). Basicamente consiste no encapsulamento e difusão dos dados (*data broadcasting*) em um fluxo de transporte junto com as *streams* de áudio e vídeo, de acordo com padrões definidos pela especificação MPEG-2 *System*, integrante da família MPEG responsável pelo processo de transmissão dos pacotes (BECKER et al, 2005; MONTEZ; BECKER, 2005).

Existem quatro mecanismos diferentes de difusão de dados: *data piping*, *data streaming*, *multiprotocol encapsulation* e mecanismo de carrosséis, sendo este último o mecanismo usualmente adotado pelos padrões de TV Digital interativa (BRACKMANN, 2008).

O carrossel é um protocolo definido pelo padrão *Digital Storage Media, Command and Control* (DSM-CC), integrante da família MPEG-2 responsável pelas funções de suporte, controle e transmissão de dados em fluxos de transporte. A função do carrossel é possibilitar a difusão de módulos de dados de forma cíclica e intercalada (periódica) para os *set-top boxes*, em um processo repetitivo. Dessa forma, quando o receptor necessitar de um determinado módulo, basta aguardar a próxima transmissão deste no ciclo de transmissão (BECKER et al, 2005; MONTEZ; BECKER, 2005). Daí a alusão a um carrossel, conforme ilustrado pela Figura 12:

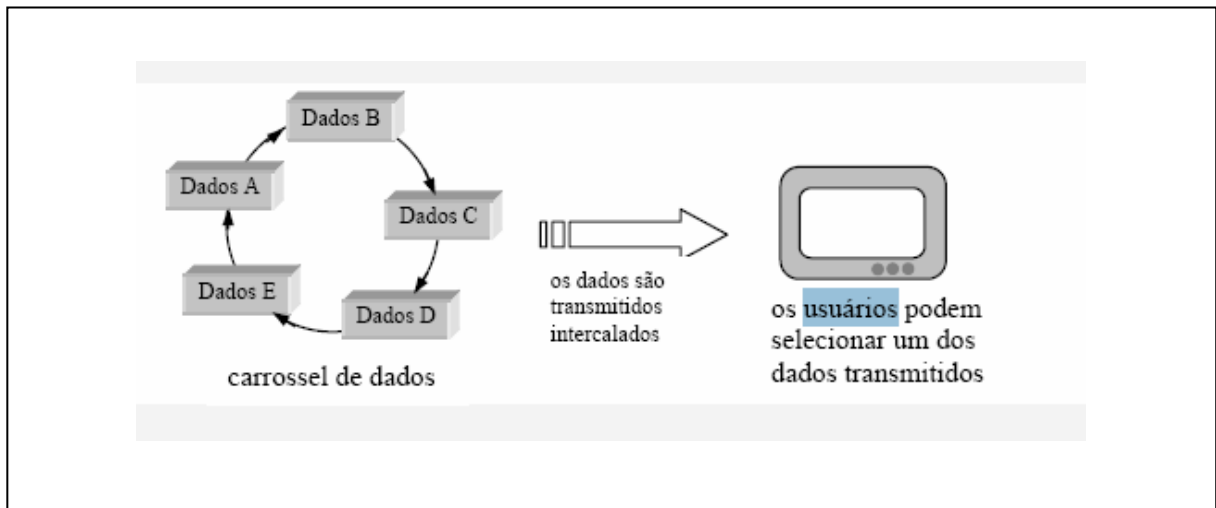


Figura 12. Funcionamento do carrossel de dados
 Fonte: MONTEZ, V.; BECKER, C. (2005, p. 84)

Os mecanismos de carrosséis costumam ser classificados em dois tipos (BECKER et al, 2005):

- a) carrossel de dados, constituído por módulos compostos de itens simples ou completos de dados (um arquivo ou conjunto de arquivos), divididos em blocos que formarão uma estrutura do tipo *Download Data Block* (DDB);
- b) carrossel de objetos: modelo mais adequado ao ambiente de TV Digital. Baseado nos carrosséis de dados, mas tratando as informações sob a forma de objetos do tipo arquivo ou do tipo diretório, componentes bases de um sistema de arquivos.

O modo carrossel de objetos permite que aplicações e objetos tais como arquivos de computador e aplicativos executáveis sejam transmitidos aos receptores e posteriormente acessados de maneira natural neste ambiente, tal qual em um computador. Além disto, caso um pacote não seja recebido adequadamente ou tenha se perdido, basta aguardar até o próximo ciclo do carrossel e capturá-lo, conforme ilustrado na Figura 13. Por estes motivos, este protocolo é a escolha natural e solução mais empregada na difusão de dados dos sistemas de TV Digital (BECKER et al, 2005; MONTEZ; BECKER, 2005).

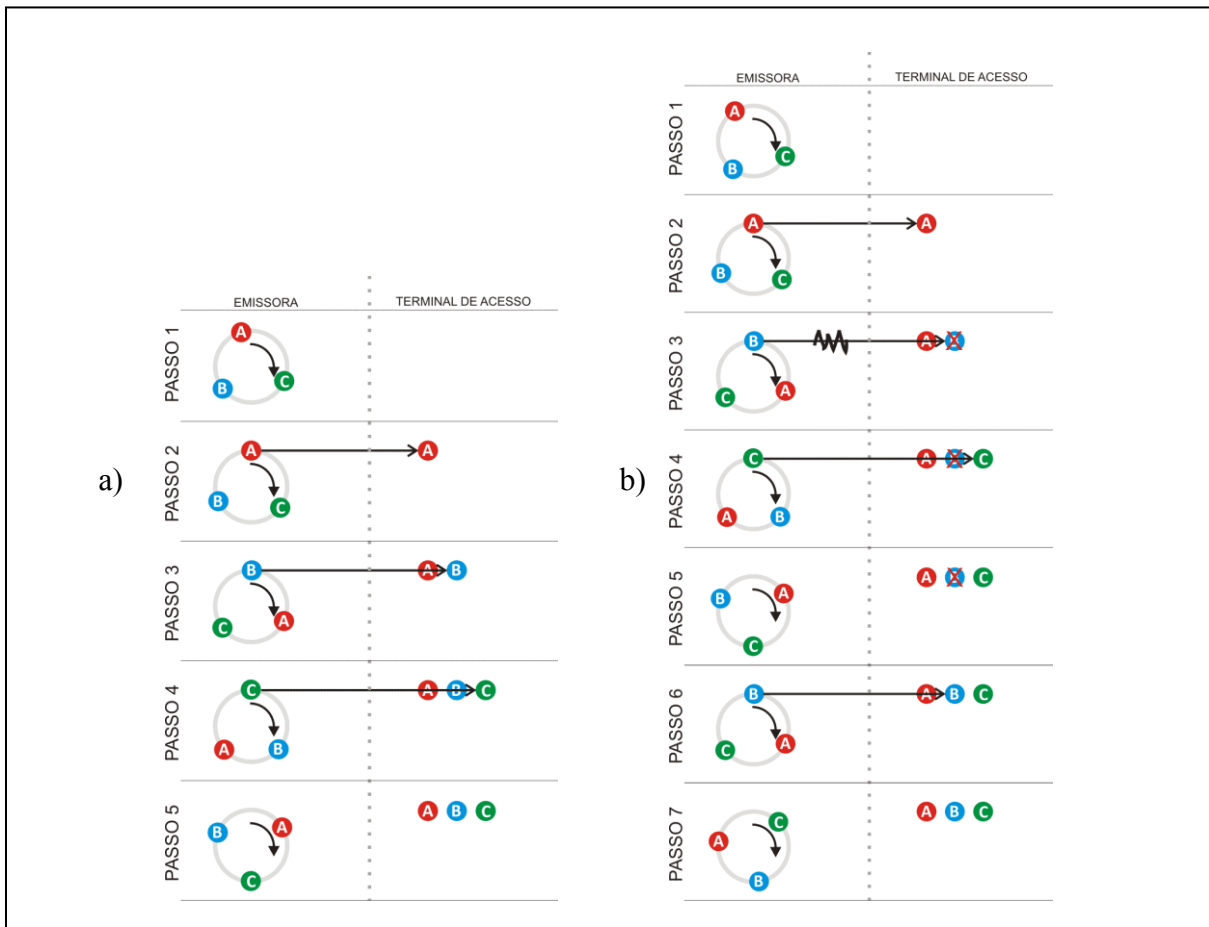


Figura 13. a) Envio via carrossel; b) Tratamento de arquivo corrompido
 Fonte: Adaptado de BRACKMANN, C. (2008, p. 48)

Segundo o grau de acoplamento entre os dados e o conteúdo audiovisual transmitido, o *datacasting* pode ser classificado em três formas: fortemente acoplado, fracamente acoplado e desacoplado. Na primeira, os dados difundidos são atrelados temporalmente (sincronizados em tempo real) ao conteúdo transmitido (uma legenda, por exemplo). Já no tipo fracamente acoplado, os dados transmitidos também são relacionados ao conteúdo televisivo, porém não estão sincronizados temporalmente ao fluxo de áudio e vídeo, como informações adicionais sobre o conteúdo transmitido; os dados então podem ser acessados pelo usuário em momento oportuno. Na modalidade desacoplada, as informações fluem entre o transmissor e o telespectador tal qual ocorre na Internet, em um processo de “navegação” do usuário em busca de dados e serviços (MONTEZ; BECKER, 2005).

2.5 O LADO DO RECEPTOR: SET-TOP BOX E CAMADA DE MIDDLEWARE

O sinal digital capturado pelas antenas *Very High Frequency/Ultra High Frequency* (VHF/UHF) ou que chega via cabo ou satélite precisa ser tratado de forma adequada por um equipamento receptor antes de ser apresentado, em uma série de etapas conforme ilustrado pela Figura 14. Este equipamento, um conjunto de componentes de hardware e software com poder computacional, pode estar embutido em um aparelho de TV Digital ou constituir um equipamento terminal de acesso à parte, chamado *set-top box*, e que serve como adaptação para que os aparelhos analógicos convencionais possam exibir o conteúdo transmitido digitalmente (BECKER et al, 2005; ETCHARTE, 2008).

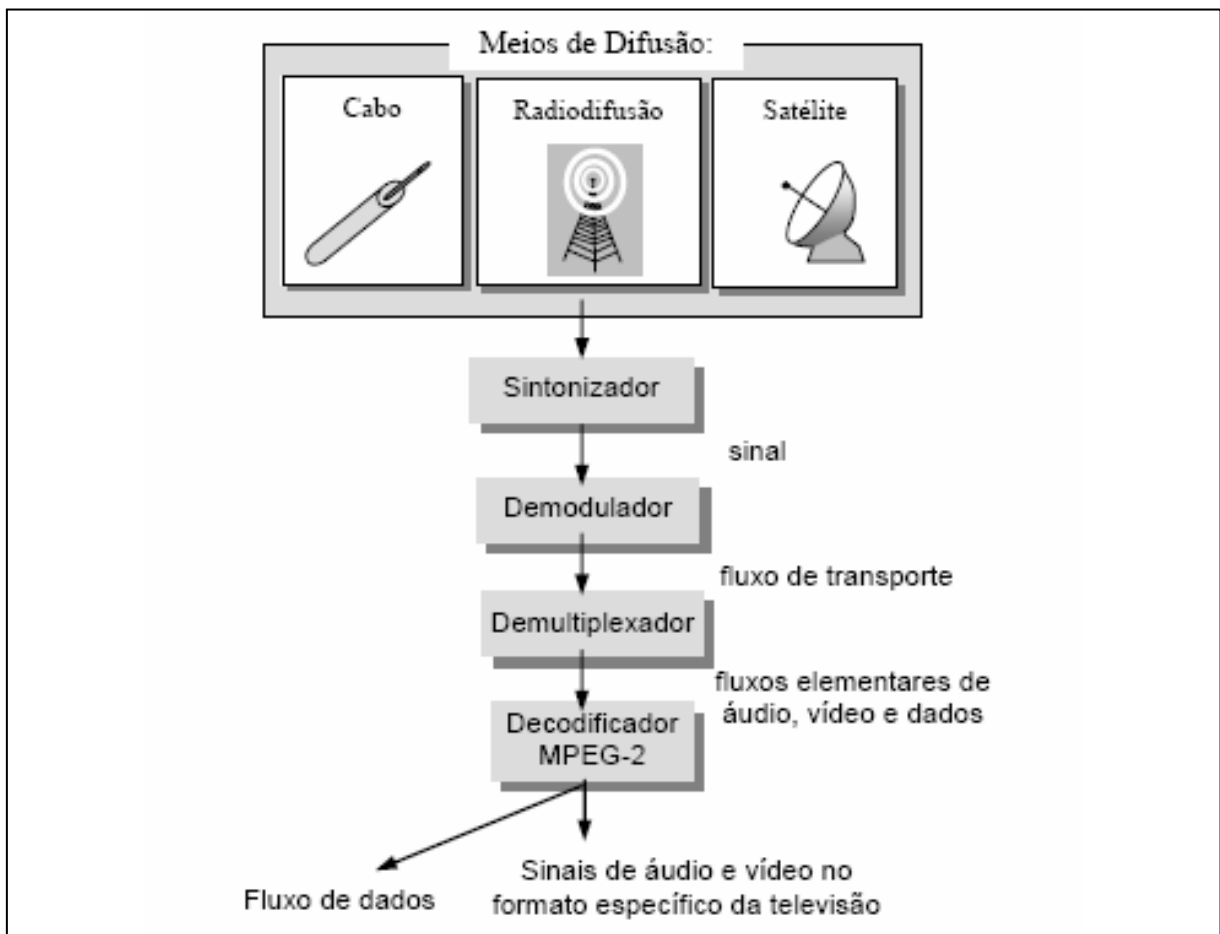


Figura 14. Etapas do processo de recepção do sinal digital
 Fonte: MONTEZ, V.; BECKER, C. (2005, p. 79)

Entre as funções exercidas pelo terminal de acesso, tem-se (BECKER et al, 2005; ETCHARTE, 2008):

- a) captura e processamento dos sinais transmitidos pelos meios difusores;
- b) demultiplexação, descompressão e decodificação dos fluxos recebidos, para que possam ser exibidos no aparelho receptor;
- c) suporte a execução de aplicações com interatividade local ou remota, via canal de retorno, quando presente;
- d) suporte à comunicação de dados bidirecional, quando presente o canal de retorno;
- e) suporte a aplicações multimídia;
- f) suporte a gravação de programas e exibição de conteúdos sob demanda;
- g) possibilitar o acesso a programas e informações disponibilizadas à plataforma de TV Digital, de forma segura e confiável, inclusive com controle de nível de acesso.



Figura 15. Exemplar de receptor *set-top box*
Fonte: ARAUJO, R. (2009, p.23)

Para que possa atender essas exigências, o *set-top box* reúne em si uma série de equipamentos de hardware, como *Central Process Unit* (CPU), unidades de armazenamento

de dados, memória principal, leitoras de cartão *smart card*⁶, e outros mais, em uma arquitetura que pode ser ilustrada pela Figura 16. Por ser um dispositivo computacional, necessita também de um sistema operacional que gerencie o funcionamento dos dispositivos de hardware, normalmente armazenado em memória *Read-Only Memory* (ROM), entre os quais podemos citar o JavaOS da Sun Microsystems, o Linux e o Microsoft Windows CE (BECKER et al, 2005; SILVA, 2003).

Além de componentes próprios, podem ser acoplados ao terminal aparelhos como *Digital Video/Versatile Disc* (DVD), mouse, câmeras, entre outros via interfaces *Institute of Electrical and Electronics Engineers* (IEEE) 1284, *Universal Serial Bus* (USB), IEEE-1394, 10 Base-T e interface serial *Recommended Standard 232* (RS-232). Inclusive conexões de rede, como mouse e ethernet (BECKER et al, 2005; SILVA, 2003).

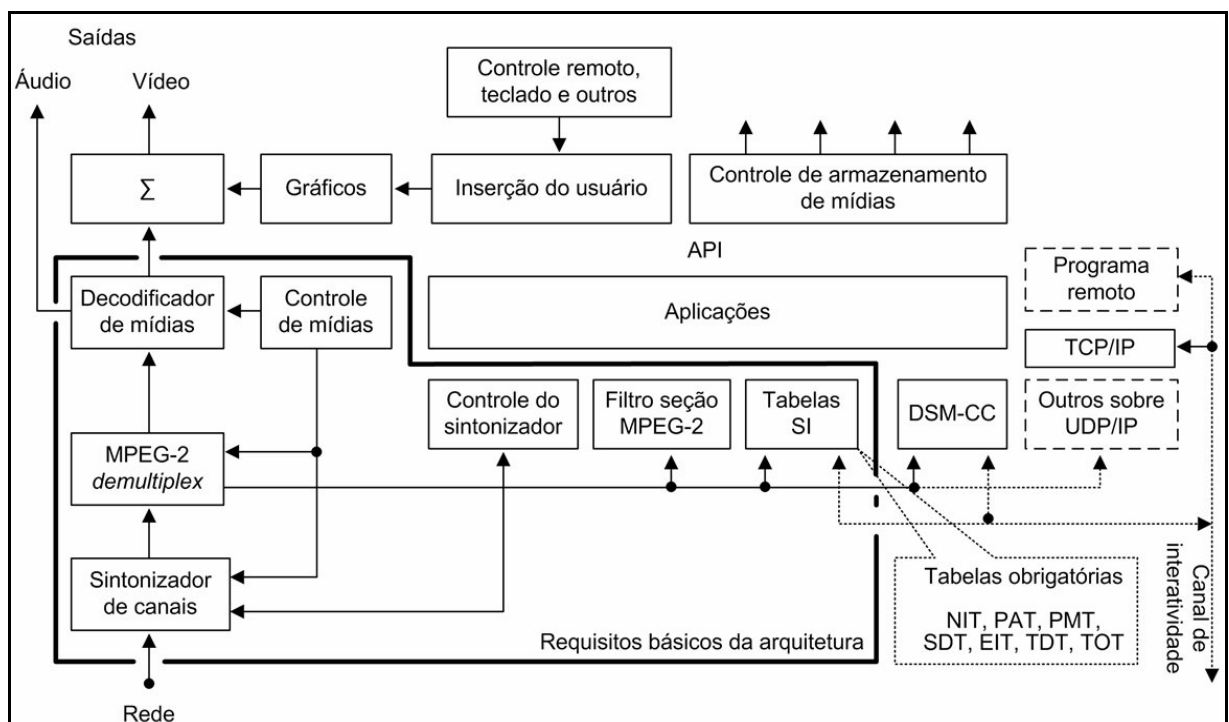


Figura 16. Arquitetura de referência do terminal de acesso
Fonte: ETCHARTE, V. (2008, p. 13)

⁶ Cartão plástico semelhante à um cartão de crédito, mas dotado de capacidade de processamento, devido à presença de um chip contendo um microprocessador e um dispositivo de memória (SMART CARD, 2010).

Remontando à arquitetura de cinco camadas discutida na seção 1.2, para que as aplicações, presentes na última camada do modelo, possam ser executadas na TV Digital interativa, se faz necessário a presença de uma camada intermediária que abstraia essas peculiaridades, denominada *middleware*. Este componente está presente nos terminais de acesso, e consiste em um conjunto de software (bibliotecas, regras e padrões) cuja finalidade é oferecer um serviço padronizado para a camada de aplicação. Esta é dispensada de conhecer os detalhes relacionados às camadas inferiores, devido a um mecanismo de abstração das heterogeneidades de redes, hardware, sistema operacional e procedimentos de demultiplexação, decodificação e descompressão (DUARTE, 2009; ETCHARTE, 2008; MONTEZ; BECKER, 2005; SOARES, 2008).

O *middleware* é uma máquina capaz de executar aplicações desenvolvidas em linguagens por ele suportadas, de forma semelhante a uma máquina virtual (DUARTE, 2009; ETCHARTE, 2008; MONTEZ; BECKER, 2005; SOARES, 2008).

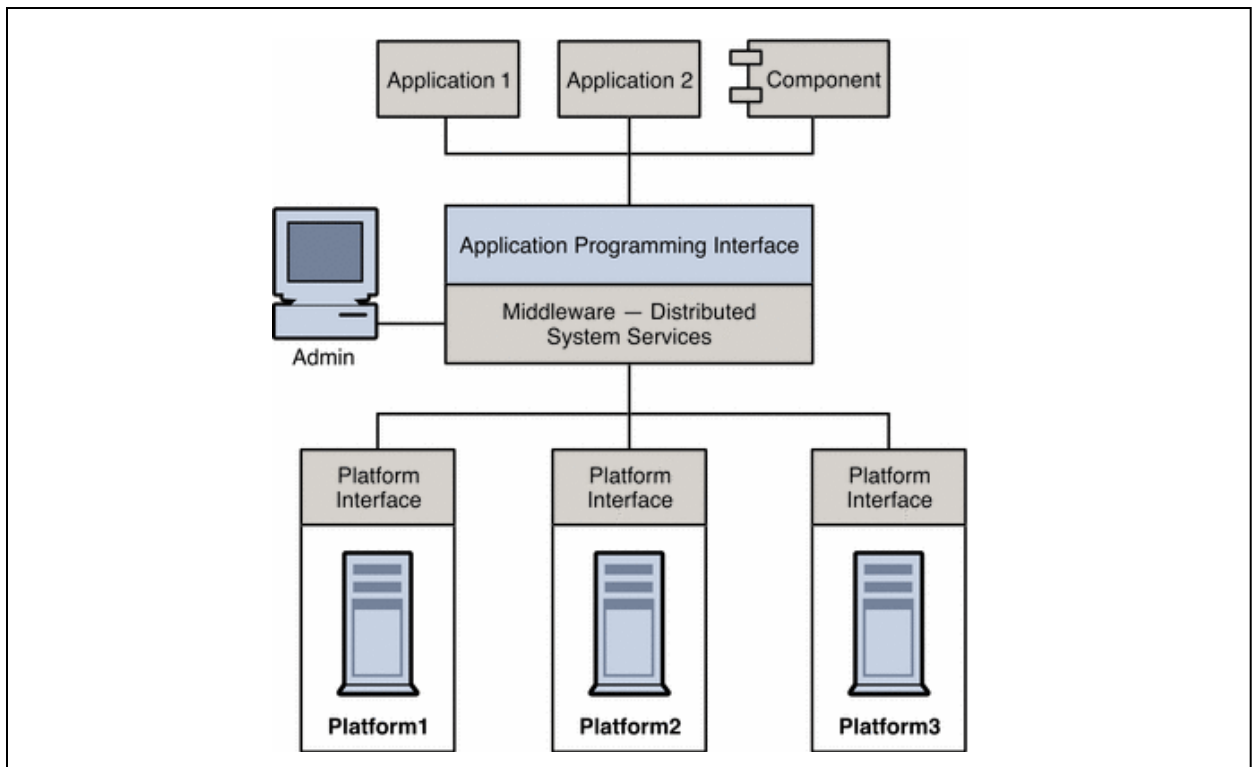


Figura 17. O *middleware* e seu papel intermediador entre as aplicações e a plataforma
 Fonte: DUARTE, D. (2009, p.29)

O emprego dos *middlewares* - prática consolidada no meio computacional e agora presente na TV Digital – tem como intuito oferecer portabilidade de aplicações entre receptores com características heterogêneas (diferentes), seja de hardware ou software, facilitando a distribuição de aplicações, interativas ou não, entre os usuários (DAMASCENO, 2008; ETCHARTE, 2008; MONTEZ; BECKER, 2005; SOARES, 2008).

Além do mais, proporciona praticidade, agilidade e eficiência no processo de desenvolvimento – o que for desenvolvido com base em determinado *middleware* será escrito uma única vez e deverá funcionar sem problemas em qualquer plataforma de receptor ou *set-top box* que empregue tal especificação, conforme ilustrado na Figura 17 (DAMASCENO, 2008; ETCHARTE, 2008; MONTEZ; BECKER, 2005; SOARES, 2008).

No processo de interatividade, o *middleware* é um dos elementos fundamentais, pois as aplicações interativas dependerão diretamente do funcionamento e serviços oferecidos por ele (DAMASCENO, 2008; ETCHARTE, 2008; MONTEZ; BECKER, 2005; SOARES, 2008).

Exemplificando de maneira prática, desde que projetada de acordo com os padrões de um mesmo *middleware*, uma mesma aplicação pode ser recebida, processada e apresentada por receptores ou *set-top boxes* tecnicamente diferentes entre si (sistema operacional, hardware, padrões de *codecs* diferentes, entre outros) devido à capacidade do *middleware* em abstrair os detalhes das demais camadas do modelo de referência genérico, considerada de fundamental importância no ambiente da TV Digital, onde as possibilidades de combinações de hardware e software diferentes são consideravelmente grandes (DAMASCENO, 2008; MONTEZ; BECKER, 2005).

Regra geral, os *middlewares* relacionados ao ambiente de televisão digital devem ser desenvolvidos obedecendo algumas exigências, entre as quais se destacam (DAMASCENO, 2008):

- a) suporte a sincronização de mídias:
 - sincronização baseada em estrutura;
 - suporte ao canal de retorno.
- b) suporte a múltiplos dispositivos de exibição;
- c) suporte ao desenvolvimento de programas em tempo de exibição – ao vivo;
- d) suporte a adaptação de conteúdo e forma como este é exibido.

Do ponto de vista de desenvolvimento, os *middlewares* da TV Digital oferecem suporte ao desenvolvimento (ambientes de desenvolvimento) e execução de aplicações (máquinas) em dois paradigmas diferentes de programação (DUARTE, 2009; ETCHARTE, 2008):

- a) declarativo: desenvolvimento utilizando linguagens declarativas ou de *script*⁷ que enfatizam a descrição declarativa e intuitiva de um problema ao invés de uma abordagem algorítmica, altamente controlada. O problema é abordado em um nível de abstração mais alto, sem envolver detalhes de programação tão aprofundados, facilitando o desenvolvimento e exigindo menos recursos de máquina. São executados pela máquina de apresentação do *middleware*;
- b) imperativo ou procedural: baseado em linguagens procedurais, onde o programador define algorítmicamente o funcionamento do programa, em um fluxo que deverá ser obedecido durante a execução. Exige mais recursos de máquina, porém oferece um maior poder de controle sobre o código ao programador, que por sua vez deverá ter capacidade técnica suficiente para tanto. No *middleware*, são executados pela máquina de Execução.

Embora haja nos *middlewares* esta divisão de ambientes paradigmáticos, há também a possibilidade de uma aplicação ser desenvolvida de maneira híbrida, combinando

⁷ Linguagens de programação executadas no interior de programas ou trechos de outras linguagens de programação (LINGUAGEM, 2010).

módulos declarativos e módulos procedurais, e isto na prática ocorre frequentemente, inclusive existindo no modelo de arquitetura, conforme se observa na Figura 18, um elemento de ponte entre os dois ambientes. Entre as linguagens declarativas mais empregadas pode-se citar a *Nested-Context Language* (NCL) e a *eXtensible HyperText Markup Language* (XHTML). Para desenvolver aplicações procedurais, normalmente utiliza-se a linguagem ou plataforma Java (DAMASCENO, 2008; ZANCANARO; SANTOS; TODESCO, 2009).

A Figura 18 ilustra o modelo de arquitetura genérica de um ambiente de execução *middleware* proposto pela *International Telecommunication Union: Telecommunication Standardization Sector* (ITU-T):

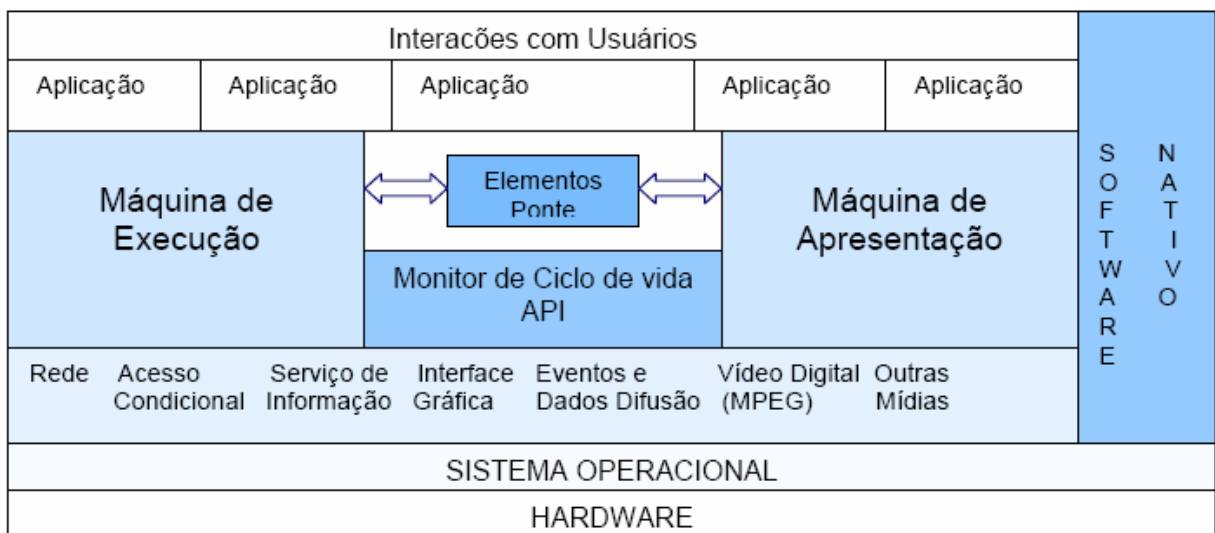


Figura 18. Arquitetura do ambiente de execução conforme ITU-T J.200

Fonte: DAMASCENO, J. (2008, p. 06)

No momento atual existem alguns padrões de TV Digital aberta em funcionamento no mundo (que serão abordadas adiante), e cada qual desenvolveu sua própria especificação de *middleware*, voltado às suas necessidades e peculiaridades, sendo eles:

- a) *Multimedia Home Platform* (MHP), desenvolvido pelo consórcio europeu *Digital Video Broadcasting* (DVB) para oferecer interatividade à TV Digital, independente de hardware e software específicos, aberto e interoperável para

receptores e *set-top boxes*. O ambiente de execução do MHP compõe-se de uma máquina virtual Java e conjunto de APIs, proporcionando maior controle sobre os recursos do receptor por parte das aplicações, além de primar pela padronização - aplicações *Digital Video Broadcasting Java* (DVB-J). A introdução da especificação MHP 1.1 possibilitou a utilização de uma linguagem de programação declarativa denominada *Digital Video Broadcasting - HyperText Markup Language* (DVB-HTML);

- b) *Digital Television (DTV) Application Software Environment* (DASE), desenvolvido pelo padrão americano *Advanced Television Systems Committee, Inc.* (ATSC). Utiliza uma máquina virtual Java, permitindo também o uso de linguagens declarativas como *HyperText Markup Language* (HTML) e *JavaScript*, de maneira semelhante ao MHP, embora ambos não sejam compatíveis. O padrão de transmissão americano está em processo de substituição do DASE pelas novas especificações *Advanced Common Application Platform* (ACAP) e *OpenCable Applications Platform* (OCAP), visando aderir ao padrão *Globally Executable MHP* (GEM), que será abordado adiante;
- c) *Association of Radio Industries and Businesses* (ARIB) desenvolvido para o padrão de transmissão ATSC japonês. Formado pelo conjunto ARIB Standard STD-B24 que define uma linguagem declarativa - *Broadcast Markup Language* (BML) – baseada no padrão *eXtensible Markup Language* (XML) e usada para especificação de serviços multimídia para TV Digital, e pela especificação ARIB-STD B23, baseada no padrão europeu (MHP) e voltada para as aplicações procedurais Java.

A Tabela 2 traça um comparativo entre os três *middlewares* retro abordados:

Tabela 2. Comparativo entre os *middlewares* MHP, DASE e ARIB

COMPARAÇÃO DOS PADRÕES DE MIDDLEWARE			
CARACTERÍSTICAS	MHP	DASE	ARIB
Segurança	SIM	SIM	Não disponível
Decodificação de conteúdo comum	SIM	SIM	SIM
Tipos de aplicativo	HTML e JavaTV	XHTML, CSS, ECMA Script, JavaTV	Não disponível
Distinção entre aplicações declarativas e procedurais	SIM	SIM	Não disponível
Interação com usuário	SIM	SIM (teclado e mouse)	SIM
Capacidade de áudio	MPEG BC	Non-streaming: (audio/basic) Streaming: (Dolby AC-3)	SIM
Capacidade de Vídeo	MPEG 2	Non-streaming: (Multiple Network Graphics) Streaming: (MPEG 2)	MPEG 2
Capacidade Gráfica	LDTV: 320 X 240 SDTV: 640 X 480 EDTV: 720 X 480 HDTV: 1920 X 1080	1920 X 1080 1280 X 720 960 X 540 640 X 480	Alta definição: 1920 X 1080; 1280 X 720 e 960 X 540. Definição Normal: 620x 480.
Display	Não disponível	Multiplano: Background, vídeo, gráfico e ponteiro/cursor (8 bit pseudo color; RGBA 4444; RGBA 5551; RGBA 6666; RGBA 8880 e RGBA 8888)	Multiplano: vídeo, figura, controle, gráfico e texto e legendas: (Y, Cr, Cb/4:2:2/8bits; Y, Cr, Cb/4:4:4/8bits/ composição do canal α em 256 valores; 1920x1080 x1 - 1 bit de controle; 8 bit para endereçamento de mapa de cores) Correção de erros sem perc.
Metadados	SIM	SIM	SIM
Receptor (STB)	Receptores Comuns de baixo custo	Receptores Comuns	Receptores Comuns de baixo custo
Extensões/Expansões	SIM	Não disponível	SIM
Serviços	HDTV, SDTV, outros serviços de telecomunicações e de dados	HDTV, SDTV, outros serviços de telecomunicações e de dados.	telecomunicações e de dados. HDTV, SDTV, outros serviços de telecomunicações e de dados.
Interatividade	SIM	SIM	SIM, via digital broadcasting, SDTV (terrestre), Satélite, redes de pacotes e redes de telecomunicações.
Controlabilidade	Funções de controle do usuário; Canais de emergência	Controle do usuário	Funções de controle do usuário; canais de emergência.
Vantagens	Baixo preço do <i>set-top box</i> ; Maior aceitabilidade mundial.	Possibilidade de contrapartidas comerciais nos EUA.	Melhor para aplicações móveis; Proximidade funcional com DVB.

Fonte: Adaptado de PAES. A; ANTONIAZZI, R. (2005, p. 27)

Além destes há o *middleware* Ginga, desenvolvido para o Sistema Brasileiro de TV Digital e aquele do padrão chinês, em fase de desenvolvimento. Vale ressaltar que todos estes citados são específicos da TV Digital aberta; entretanto, podemos encontrar no mercado alternativas proprietárias, dentre os quais citamos: Microsoft/Alcatel[®] TV, Lucent's MiView TV[®], Bstream *Middleware*, Alticator[®], entre outros (CRUZ; MORENO; SOARES, 2008).

Em virtude de diferenças na estrutura das bibliotecas de *software* existentes entre os *middlewares* abertos MHP, DASE e ARIB, aplicações escritas em determinado padrão não eram compatíveis com o ambiente dos demais, de início. Neste sentido, o grupo DVB propôs a criação do *framework* GEM, um padrão de harmonização com objetivo de permitir, a princípio, que aplicações desenvolvidas para o *middleware* do padrão europeu pudessem ser executadas nos ambientes dos padrões americano e japonês sem problemas (ANDREATA, 2006; CARVALHO, 2008; DAMASCENO, 2008; GUIMARÃES; COSTA, 2006).

O GEM não é considerado como uma especificação de *middleware*, mas um padrão contendo partes da *Application Programming Interface* (API) procedural do MHP compatíveis com os demais sistemas (capturando as interfaces e semântica do MHP independentes da plataforma DVB), conjuntos de protocolos, especificações de formato de conteúdo e componentes de suporte às necessidades inerentes aos outros padrões (americano e japonês), promovendo assim a interoperabilidade entre os padrões aderentes (ANDREATA, 2006; CARVALHO, 2008; DAMASCENO, 2008; GUIMARÃES; COSTA, 2006).

Desta forma, garante-se que uma aplicação procedural baseada no núcleo definido pelo GEM possa ser executada em ambientes de outros padrões, desde que aderentes ao consórcio (ANDREATA, 2006; CARVALHO, 2008; DAMASCENO, 2008; GUIMARÃES; COSTA, 2006). O GEM pode ser visto como uma interseção entre os padrões internacionais que o adotam, formando uma estrutura interoperável conforme ilustrado na Figura 19:

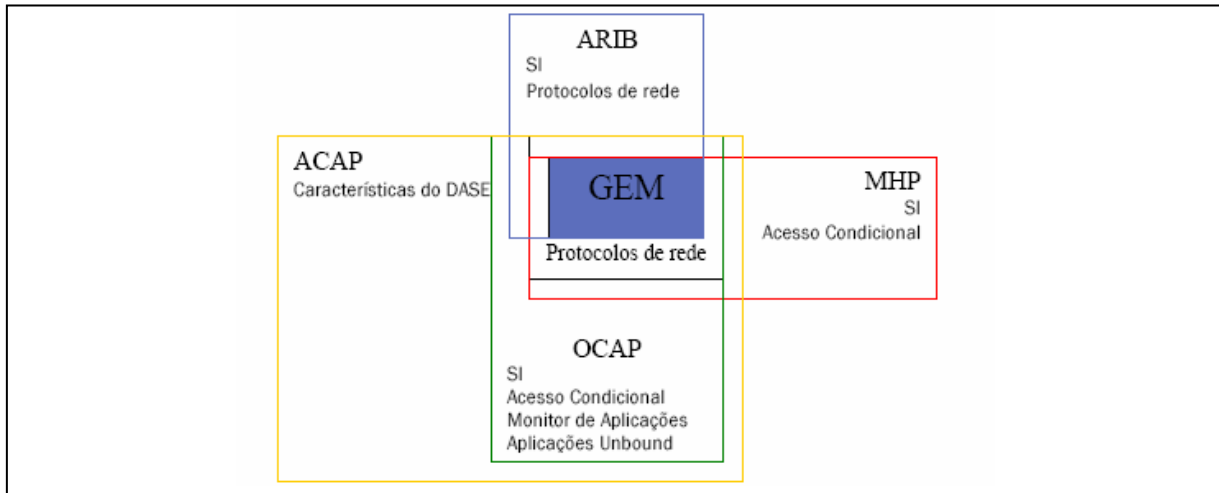


Figura 19. O GEM e a relação com os *middlewares* dos padrões de TV Digital aberta
 Fonte: DAMASCENO, J. (2008, p. 16)

2.6 SISTEMAS PIONEIROS DE TV DIGITAL

As pesquisas em torno da digitalização do sistema televisivo foram iniciadas pelo Japão, nos anos 70, seguido pela Europa e Estados Unidos, resultando no nascimento dos respectivos padrões ou sistemas de TV Digital aberta, considerados como os três principais em funcionamento atualmente (BECKER; MONTEZ, 2004; CASTRO; CODATO, 2008; DARÓS, 2004).

O padrão americano ATSC iniciou suas transmissões em 1998, adotado posteriormente por países como Canadá, Coréia do Sul, Taiwan e México (ANDREATA, 2006; LOPES, 2007; MONTEIRO, 2009). Tem por prioridade a transmissão em alta definição (HDTV), não oferecendo suporte à interatividade. Com relação à transmissão terrestre, caracteriza-se pela baixa robustez de seu sinal, apresentando alta sensibilidade a interferências, problemas na recepção em antenas internas e recepção móvel/portátil inviável. Isto ocorre devido à adoção do padrão de modulação 8-VSB, que funciona muito bem em transmissão via cabo, muito popular naquele país, mas deixa a desejar quando aplicado em difusão terrestre (ALMEIDA JUNIOR, 2008; ANDREATA, 2006; ARNOLDO, 2008;

LOPES, 2007; PAES; ANTONIAZZI, 2005). A Figura 20 apresenta o diagrama de blocos do sistema ATSC:

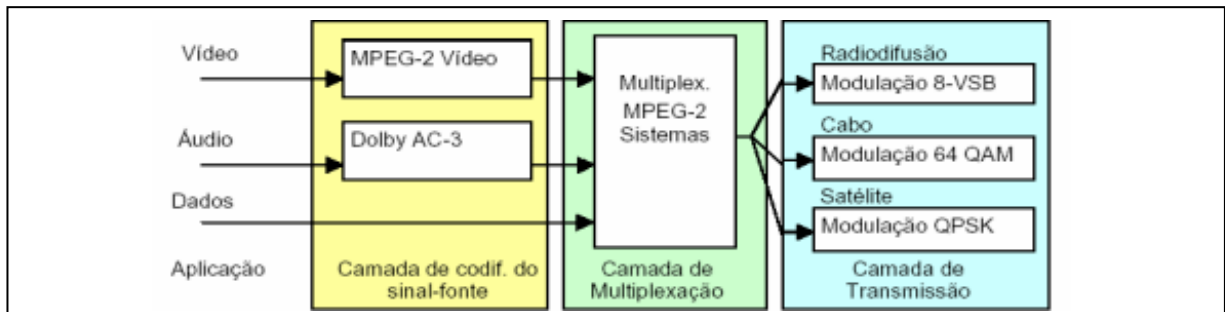


Figura 20. Diagrama de blocos do sistema ATSC
Fonte: PAES. A; ANTONIAZZI, R. (2005, p. 06)

O padrão europeu DVB iniciou as transmissões no meio terrestre em 1998, sendo atualmente o padrão adotado pela maioria dos países do mundo que já adotaram a TV Digital, até o momento (ANDREATA, 2006; BECKER; MONTEZ, 2004; DARÓS, 2004; MONTEZ; BECKER, 2005). Diferentemente do padrão americano, o europeu se difere pela robustez no sinal transmitido, com alta imunidade a interferências e bom desempenho na recepção interna, graças à adoção da modulação COFDM (ANDREATA, 2006; DARÓS, 2004; PAES; ANTONIAZZI, 2005).

Optou-se também por oferecer uma programação mais rica em detrimento da qualidade de resolução, primando-se pela multiprogramação (até seis programas simultâneos por canal), qualidade de transmissão hierárquica (permitindo a recepção simultânea em diversos níveis de resolução, de acordo com a capacidade do receptor) e o suporte a serviços interativos, contando inclusive com a colaboração de empresas de telefonia celular no desenvolvimento de um canal de retorno em tecnologia *Global System for Mobile Communication* (GSM), como meio alternativo para prover a interatividade (ANDREATA, 2006; DARÓS, 2004; FERREIRA, 2008; GUIMARÃES; COSTA, 2006; PAES; ANTONIAZZI, 2005). A Figura 21 apresenta o diagrama de blocos do sistema DVB:

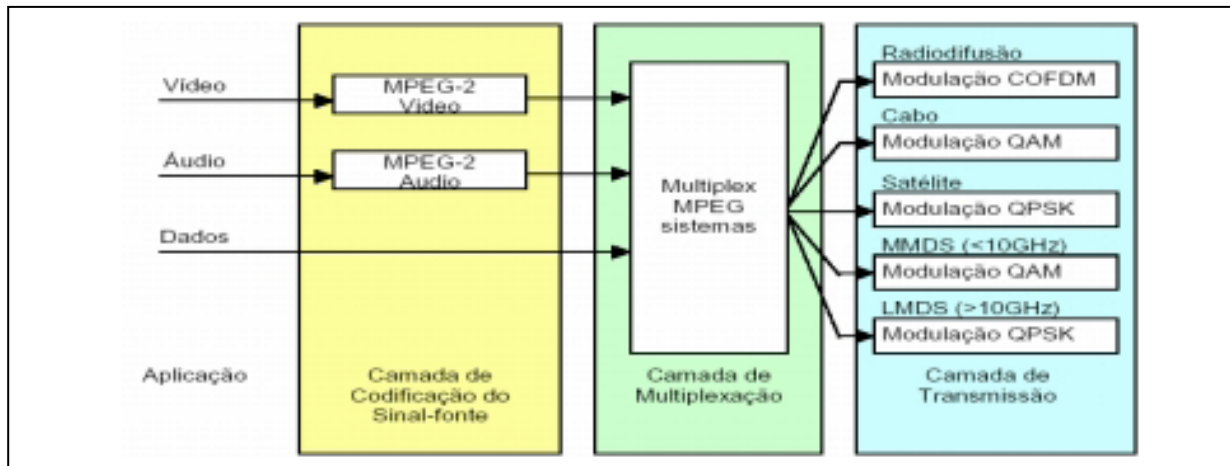


Figura 21. Diagrama de blocos da estrutura do sistema DVB
 Fonte: PAES, A.; ANTONIAZZI, R. (2005, p. 07)

O padrão japonês ISDB foi o último dos três a entrar em operação, sendo em 2001 via satélite e em 2003 via difusão terrestre, embora este país seja o pioneiro nas pesquisas em TV Digital (ANDREATA, 2006; FERREIRA, 2008).

Sendo considerado como uma evolução do sistema europeu, destaca-se pela robustez ainda maior na transmissão, oferecendo qualidade de recepção até mesmo a dispositivos móveis e portáteis, caracterizando-o como o mais flexível dentre os três sistemas vistos (ALVES, 2008; ANDREATA, 2006; DARÓS, 2004; FERREIRA, 2008; PAES; ANTONIAZZI, 2005).

Prioriza a transmissão em HDTV, oferecendo suporte a transmissão hierárquica, assim como o modelo europeu, possibilitando a multiprogramação de canal e recepção em modos de resolução e qualidade diferenciados, como STDV e LDTV (ANDREATA, 2006; AYRES, 2007; FERREIRA, 2008), assim como prioriza a oferta de recursos de interatividade, multimídia e *datacasting*, incluindo a disponibilização de um canal de interatividade via *Asymmetric Digital Subscriber Line* (ADSL), trabalho conjunto com o sistema de telefonia e Internet (FERREIRA, 2008; LOPES, 2007).

A Figura 22 apresenta o diagrama de blocos do sistema ISDB:

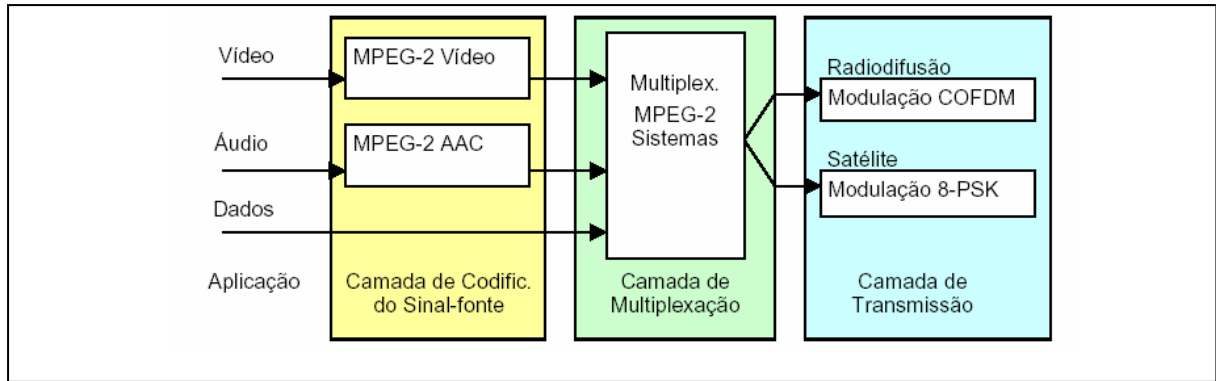


Figura 22. Diagrama de blocos do sistema ISDB
 Fonte: PAES. A; ANTONIAZZI, R. (2005, p. 09)

Apesar das diferenças elencadas, os três sistemas demonstram similaridade em algumas características técnicas, como nos mostra o comparativo a seguir:

Tabela 3. Comparativo entre os padrões americano, europeu e japonês

CARACTERÍSTICAS	ATSC	DVB-T	ISDB-T
Modulação	8-VSB	COFDM (QPSK, 16QAM e 64 QAM)	COFDM (QPSK, 16QAM e 64 QAM)
Multiplexação	MPEG-2	MPEG-2	MPEG-2
Codificação de áudio	Dolby AC-3	MPEG-2 Layer II audio ou Dolby AC-3	MPEG-2 AAC audio
Codificação de vídeo	MPEG-2	MPEG-2	MPEG-2
Resolução	Alta definição (HDTV)	Definições simultâneas (prioridade para SDTV)	Definições simultâneas (prioridade para HDTV)
Canalização	6 MHz (escalonável para 7 ou 8 MHz)	7 ou 8 MHz (pode ser escalonado para 6 MHz)	6 MHz (escalonável para 7 ou 8 MHz)
Taxa de transmissão	19,4 Mbit/s	4,98~31,67 Mbit/s	3,65~23,23 Mbit/s
Middleware	DASE	MHP	ARIB-Std-B24

Fonte: Adaptado de ALMEIDA JUNIOR, E. (2008, p. 11); ANDREATA, J. (2006, p. 17-20)

Além dos três padrões apresentados e do padrão brasileiro que será visto no Capítulo 6, a China também vem desenvolvendo trabalhos com a intenção de definir um padrão nacional de TV Digital. No início, cinco propostas foram oferecidas (ALMEIDA JUNIOR, 2008; PAES; ANTONIAZZI, 2005), sendo que o padrão adotado pelo governo chinês é o *Digital Multimedia Broadcasting Terrestrial/Handhelding* (DMB-T/H), resultado de pesquisas conduzidas pelas universidades *Tsinghua University* de *Beijing* (Pequim), e *Jiaotong University*, de Shanghai. Estas instituições desenvolveram padrões separados, que foram unidos, passando a coexistir em prol da instituição de um padrão que atendesse às necessidades chinesas, tendo como base tecnologias herdadas dos padrões europeu, japonês e americano, visando oferecer um sistema de alta definição (HDTV) e possibilidade de recepção móvel em um mesmo canal (BURGER et al., 2007; PAES; ANTONIAZZI, 2005).

Compreendidas as questões de ordem técnica relacionadas à estrutura de um sistema de televisão digital e apresentados os três padrões pioneiros, a próxima etapa é abordar questão da interatividade no contexto da televisão digital, suas classificações e possibilidades.

3 A INTERATIVIDADE NA TV DIGITAL

Nos últimos tempos tem-se ouvido com frequência cada vez maior o termo *interatividade*. Produtos, serviços e entretenimento são vendidos como que dotados de “interatividade”, em uma estratégia puramente de marketing, sendo que muitas vezes o que ocorre na realidade é a oferta de interação ou melhorias neste processo. Este emprego errôneo protagonizado pelos meios publicitários, confundindo interação e interatividade, desencadeou um processo de banalização do conceito, sendo aplicado a qualquer produto ou serviço que ofereça algum grau de participação – real ou suposta – do usuário (LEMOS, 1997; MONTEZ; BECKER, 2005; PASCHOAL NETO; CARVALHO, 2008).

Neste sentido, em um primeiro momento é necessário que se diferencie interação e interatividade. Segundo Montez e Becker (2005) a interação é inerente ao ser humano e pode ocorrer diretamente entre dois ou mais entes, ao passo que a interatividade necessita de um meio eletrônico intermediário entre as partes, sejam elas pessoas, sistemas ou máquinas.

Embora o conceito de interatividade venha sendo discutido sob diferentes óticas científicas e filosóficas, sob o aspecto tecnológico a interatividade se caracteriza como uma nova forma de interação homem-máquina, na qual via dispositivos e interfaces avançadas o usuário pode interferir na produção e apresentação do conteúdo apresentado, em uma comunicação bidirecional, onde se pode ser emissor, produtor e consumidor, ao mesmo tempo (BREUNIG; MARQUES, 2008; MONTEZ; BECKER, 2005; LEMOS, 1997).

Montez e Becker (2005), definindo a interatividade de um processo ou ação como uma atividade mútua e simultânea entre agentes, envolvidos na busca de um objetivo comum, citam as características definidas por Lippman como necessárias para que um sistema possa ser considerado de fato interativo (BREUNIG; MARQUES, 2008; MONTEZ; BECKER, 2005):

- a) interruptabilidade: os participantes devem ter autonomia, podendo interromper o processo no momento em que bem entenderem;
- b) granularidade: refere-se ao menor elemento após o qual se pode interromper o processo, e à resposta dada a esta interrupção do fluxo. O sistema não pode deixar dúvidas de que o pedido foi entendido, evitando passar a impressão ao usuário de que o sistema está travado;
- c) degradação suave: refere-se ao comportamento do sistema quando este não tem resposta para dar ao usuário. O sistema não pode se desligar, nem deixar o usuário sem resposta;
- d) previsão limitada: embora seja difícil programar todas as alternativas possíveis, o sistema deve prever ao máximo as possibilidades de ocorrências, e caso ocorra algo imprevisto na relação de interação, o sistema deve ser capaz de lidar com esta situação, sempre oferecendo alguma resposta ao usuário;
- e) não *default*: o sistema não pode induzir os participantes a agirem de forma determinada, proporcionando liberdade ao usuário para interagir da forma como melhor lhe aprouver.

Nessa esteira, a interatividade na televisão representa uma quebra de paradigma. O usuário deixa de ser passivo ou no máximo reativo - condição característica da TV tradicional - para transformar-se em agente participativo, interferindo de fato na produção ou exibição dos conteúdos que serão transmitidos de várias maneiras como: escolha de ângulos de transmissão, solicitação de aplicações ou programas específicos, intervenção na grade de programação das emissoras e, em um grau mais alto de interatividade, atuando como produtor, enviando ao emissor conteúdo de sua própria autoria (CASTRO; CODATO, 2008; MONTEIRO, 2002; MONTEZ; BECKER, 2005; PASCHOAL NETO; CARVALHO, 2008).

A TV Digital interativa surge como uma nova mídia, onde a interatividade é

possibilitada pelo que se chama de “convergência tecnológica”, caracterizada pela agregação de diferentes tecnologias e meios de comunicação (informática, telefonia celular, Internet e outros mais) ao sistema televisivo, modificando não só atitude do telespectador como também o processo de produção das emissoras (BECKER; MONTEZ, 2004; MACKLIN, 2002; MÉDOLA; TEIXEIRA, 2007; MONTEIRO, 2002).

Os principais elementos necessários à interatividade na TV Digital são: a digitalização do processo de transmissão (multiplexação), os geradores de carrossel, receptores dotados de capacidade computacional e um canal de retorno, proporcionando a comunicação entre o telespectador e o emissor (BREUNIG; MARQUES, 2008; MATOS; JULIÃO; MONTEIRO, 2002; SANTOS, 2007).

3.1 NÍVEIS DE INTERATIVIDADE NA TELEVISÃO

De acordo com Lemos (1997), a interatividade no ambiente televisivo do ponto de vista de sua evolução através dos tempos, pode ser classificada em cinco níveis (BREUNIG; MARQUES, 2008; LEMOS, 1997; MONTEZ; BECKER, 2005):

- a) 0: televisão em preto e branco e com poucas opções de canal, onde a interação com o usuário se limitava à troca de canais, ligar e desligar e fazer ajustes como brilho e contraste;
- b) 1: televisão em cores e aumento do número de emissoras. Controle remoto surge como instrumento que facilita a navegação entre os canais;
- c) 2: caracterizado pelo surgimento de equipamentos acopláveis à televisão, como videocassete, aparelhos de videogame e câmeras portáteis. O usuário pode agora gravar conteúdos, e assistir estes ou conteúdos externos (adquiridos em videolocadoras, por exemplo) em momento oportuno;

- d) 3: surgimento dos primeiros mecanismos com características interativas, como enquetes respondidas via telefone ou email. O telespectador passou a interagir de forma reativa, respondendo a um estímulo gerado pela emissora;
- e) 4: para Lemos (1997) é onde se estabelece a interatividade na televisão, exercida por mecanismos como escolha de ângulos de transmissão ou conteúdos transmitidos, proporcionando ao telespectador uma maior interação com o conteúdo transmitido em tempo real.

Embora Lemos (1997) considere o nível 4 como ápice do processo de interatividade na televisão, Montez e Becker (2005) discordam no sentido de que neste nível a interatividade ainda é muito limitada, resumindo-se à escolha de conteúdos transmitidos ou respostas a estímulos enviados pelo transmissor, classificando a TV como reativa até este estágio. Neste sentido, propuseram a inclusão de mais três níveis necessários para transformar o telespectador em usuário pró-ativo (MONTEZ; BECKER, 2005):

- f) 5: presença mais efetiva do telespectador no conteúdo, fugindo do papel restrito a receber conteúdos pré-definidos pelo transmissor. O usuário agora pode participar da programação enviando vídeos de baixa qualidade de sua própria autoria;
- g) 6: possibilidade de envio de vídeos de alta qualidade produzidos pelo usuário, devido ao aumento de banda no canal de interatividade;
- h) 7: atinge a plenitude na interatividade, onde o usuário telespectador se confunde com o transmissor, no sentido de que passa a ser também gerador de conteúdo, caso disponha de ferramentas adequadas à produção de programas, de forma semelhante à febre dos *blogs*⁸, *fotologs*⁹ e páginas na Internet. O modelo de produção e veiculação do conteúdo televisivo é rompido.

⁸ Contração do termo *web log*: página na Internet cuja estrutura permite a atualização rápida a partir de artigos ou “posts”, organizados em ordem cronológica por data de postagem (BLOG, 2010).

Percebe-se que para que a interatividade aconteça a partir do nível 5, se faz necessária a presença de um canal de retorno, possibilitando a comunicação inversa à difusão, interligando as duas extremidades (emissor e receptor), no sentido receptor – emissor. A transição entre os níveis 4 e 5 representa uma quebra de paradigmas, onde a comunicação deixa de ser unidirecional e o telespectador rompe a barreira de passividade (BREUNIG; MARQUES, 2008; CRUZ, 2009; MONTEZ; BECKER, 2005).

Sob o ponto de vista do nível de uso do canal de interatividade, o Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) sugere a classificação da interatividade em três níveis distintos, no contexto dos projetos relacionados ao SBTVD (DALL'ANTONIA, 2004; TAVARES et al, 2007): local, intermitente e permanente, conforme apresentado na Tabela 4:

Tabela 4. Classificação dos níveis de interatividade em função do canal de retorno

INTERATIVIDADE	CANAL DE RETORNO	NOVOS SERVIÇOS
Local	Ausente	Guia de programação, extras, informações adicionais, downloads de aplicativos, datacasting, multicâmera (em SDTV)
Intermitente	Comunicação assíncrona (não é tempo real)	Votação, correio eletrônico, comércio eletrônico pela TV, T-Gov, downloads sob demanda
Permanente	Comunicação síncrona (em tempo real)	Mensagens instantâneas, jogos on-line, serviços bancários, acesso à Internet.

Fonte: TAVARES, T. et al (2007, p. 36)

A modalidade local é a mais básica, onde o usuário não estabelece comunicação com o emissor via canal de retorno; limita-se ao controle de aplicações residentes no aparelho receptor, ou enviadas pelo emissor via difusão. Na modalidade intermitente, já se faz presente

⁹ Também conhecido como *flog*. De estrutura semelhante ao blog, mas veiculando fotos (FLOG, 2010).

o canal de retorno, em sentido unidirecional (dados trafegam em uma direção por vez) e não dedicado (não disponível em tempo integral), de forma assíncrona. Já a interatividade permanente representa o grau mais avançado, onde uma comunicação bidirecional e síncrona é estabelecida entre o receptor e o emissor, de forma dedicada. Também é possível a interligação com provedores de conteúdo e até mesmo com outros usuários (CRUZ, 2009; MATOS; JULIÃO; SANTOS, 2007).

Quanto à interferência do usuário sobre o conteúdo, os níveis de interatividade podem ser classificados em três níveis distintos (BREUNIG; MARQUES, 2008; KUNG; KENZO; ZANINI, 2007; MONTEZ; BECKER, 2005; REISMAN, 2002):

- a) reativo: opções de conteúdo e resposta definidos no programa, restando ao usuário apenas escolher o que deseja acessar. Alguns exemplos são: seleção de legendas, canais e exibição de informações adicionais em uma cena;
- b) coativo: o usuário pode controlar a sequência, ritmo e estilo de execução, interagindo com as aplicações de maneira local (interatividade local). Alguns exemplos: jogos interativos, propagandas interativas com seleção de cenas e aplicações “*quiz*”¹⁰;
- c) proativo: neste nível, o usuário pode controlar tanto a estrutura como o conteúdo, atingindo o grau de interatividade remota, onde é possível solicitar conteúdos ao emissor via canal de retorno, enviar dados e conteúdos de autoria própria a ele e também acessar conteúdos e serviços disponíveis na Internet.

Segundo Kung, Kenzo e Zanini (2007) os produtores de conteúdo poderão desenvolver conteúdos mais ricos e atrativos, que interajam com o telespectador e prendam a sua atenção, servindo então como instrumento em potencial de *merchandising*.

¹⁰ Forma de avaliar simultaneamente uma grande quantidade de pessoas, utilizando um questionário do tipo “certo” ou “errado” (QUIZ, 2010).

3.2 CLASSIFICAÇÃO DOS CONTEÚDOS E APLICAÇÕES INTERATIVAS

De acordo com a instituição americana Emarketer, o termo TV interativa engloba uma série de aplicações, serviços e tecnologias, algumas já existentes e outras que ainda serão desenvolvidas. Neste sentido, a instituição estabeleceu uma classificação destes conteúdo em sete grupos distintos (BREUNIG; MARQUES, 2008; MACKLIN, 2002; MONTEZ; BECKER, 2005):

- a) TV avançada (*enhanced tv*): conteúdo envolvendo texto, vídeo e elementos gráficos, como fotos e animações, apresentados de forma integrada e organizados em uma grade de programação – Figura 23 (a);
- b) Internet na TV: acesso à Internet e todos os seus recursos. Aplicações como *t-commerce* fazem parte deste grupo – vide Figura 23 (b);



Figura 23. a) aplicação *enhanced tv* da rede CNN; b) aplicação *t-commerce*
 Fonte: CNN, 2006; LOWTRUX, 2009

- c) TV individualizada: permite a adaptação dos conteúdos conforme o gosto do usuário (ângulos de transmissão, fontes, cores), como também a repetição de cenas perdidas, entre outros;

- d) vídeo sob demanda: possibilidade de o telespectador assistir a programas da grade da emissora no horário em que desejar (com exceção dos programas ao vivo), desde que a emissora ofereça este serviço;
- e) *Personal Video Recorder* (PVR): permite a gravação de conteúdos digitais em alta definição, bastando apenas que o usuário especifique o título, horário, assunto ou qualquer informação suficiente para identificar o conteúdo a ser gravado. O programa então é gravado no disco rígido do receptor;
- f) *Walled Garden*: portal de conteúdo instrutivo, contendo um guia das aplicações interativas, esclarecendo o usuário quanto às possibilidades e opções de aplicações disponíveis;
- g) console de jogos: utilização da televisão em jogos eletrônicos, podendo ter como adversários outros usuários o próprio receptor ou até mesmo outros usuários, em rede.

Montez e Becker (2005), mencionando Gawlisnki, sugeriram a inclusão de mais dois grupos, para que se possa cobrir todo o contexto de aplicações e serviços possibilitados pela TV Digital interativa (BREUNIG; MARQUES, 2008; MONTEZ; BECKER, 2005):

- h) Guia de Programação Eletrônica (EPG): portal contendo um guia de programação da emissora, semelhante aos impressos distribuídos pelas TV's por assinatura. Permite ao usuário selecionar a programação desejada, usando o controle remoto; entretanto, não deve ser confundido com as aplicações classificadas como do grupo *walled garden*, que consiste em um guia de aplicações interativas disponíveis para instalação ou execução no receptor em determinado momento. A Figura 24 mostra um exemplo de aplicação EPG:



Figura 24. Aplicação de guia de programação eletrônica
 Fonte: BOQUIMPANI, A. (2009, p. 26)

- i) serviços de teletexto: informações adicionais sobre a programação, como previsão do tempo, informações econômicas, em forma de texto, podendo se sobrepôr ao conteúdo televisivo exibido ou ocupar a tela inteira.

3.3 O CANAL DE RETORNO OU DE INTERATIVIDADE

O canal de retorno ou de interatividade é o sistema ou meio que permite a interação individual entre o usuário e o emissor, encaminhando solicitações e informações e, no nível mais alto de interatividade, enviando ao emissor conteúdo de autoria própria ou recebendo conteúdos solicitados sob demanda (LAMAS et al, 2005; MONTEZ; BECKER, 2005). Possibilitando a interatividade intermitente ou permanente, segundo o conceito do CPqD, ou então a partir do nível 5, de acordo com Montez e Becker (2005) o canal de retorno é o elemento que possibilita a transformação do papel do telespectador, deixando de ser agente passivo ou reativo para tornar-se usuário de fato, proporcionando uma ligação direta entre o usuário e os provedores de conteúdo (BREUNIG; MARQUES, 2008; ETCHARTE, 2008; LAMAS et al, 2005; MONTEZ; BECKER, 2005).

No contexto da TV Digital por difusão terrestre, o sistema de canal de

interatividade é resultado da interconexão das redes de transmissão televisivas com a rede de telecomunicações e seu hibridismo de tecnologias. Esta convergência tecnológica resulta em um sistema de comunicação bidirecional, onde o meio de difusão terrestre funciona como um canal de descida (entre o emissor e o receptor) e a malha de telecomunicação funcionando como canal de retorno entre o usuário/receptor e o emissor/provedores de conteúdo, ilustrados na Figura 25 (ETCHARTE, 2008; LAMAS et al, 2005).

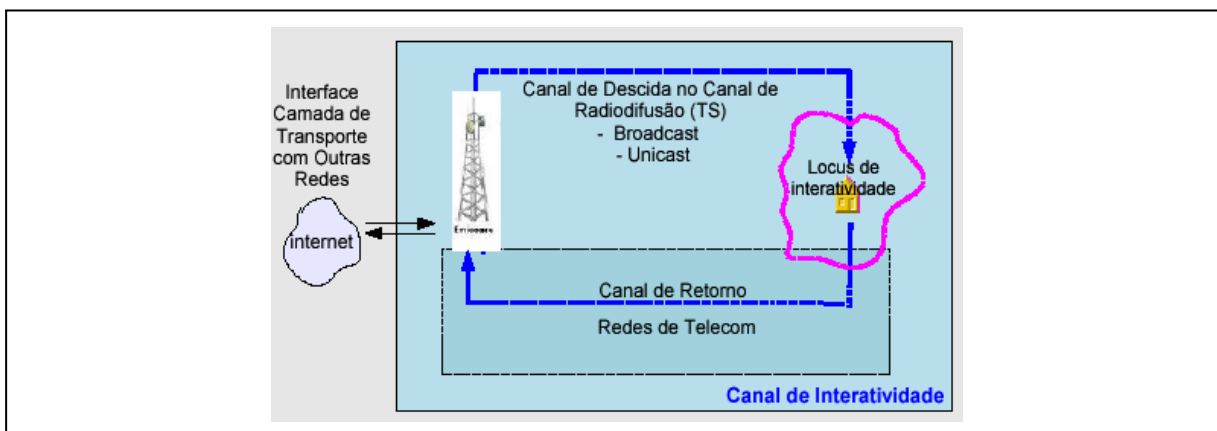


Figura 25. Diagrama simplificado do canal de interatividade
Fonte: ETCHARTE, V. (2008, p. 04)

Enquanto a tecnologia que proporciona o canal de descida é inerente ao sistema televisivo – correspondendo ao meio de transmissão, no caso a difusão terrestre - o canal de interatividade por sua vez é implementado com base em tecnologias que não pertencem ao meio televisivo e sim às redes de comunicações disponíveis (ETCHARTE, 2008; LAMAS et al, 2005).

Esta dependência dos meios de telecomunicação constitui uma barreira a ser superada no processo de implantação da TV interativa, considerada pelo governo como instrumento de inclusão digital visto à grande penetração da televisão junto às camadas mais baixas. A implantação de um sistema de canal de retorno eficiente e viável ainda não se concretizou mesmo nos países onde a TV Digital aberta não é mais novidade há mais de uma

década. No contexto brasileiro, onde por um lado a cobertura da televisão aberta é superior a 90% da população (pelo fato de ser gratuita), de outra banda o acesso aos meios de telecomunicação e tecnologias digitais ainda é restrito, devido a questões econômicas e geográficas (BASTOS, 2008; LAMAS et al, 2005; MONTEZ; BECKER, 2005; TABORDA et al, 2007).

A Tabela 5 ilustra a distribuição do acesso doméstico à Internet no Brasil, resultado de uma pesquisa realizada pelo Centro Brasileiro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETICbr) entre setembro e novembro de 2008

Tabela 5. Proporção de domicílios com acesso à Internet no Brasil

PORCENTAGEM (%)		SIM	NÃO
TOTAL BRASIL		18	82
ÁREA	URBANA	20	80
	RURAL	4	96
REGIÕES DO PAÍS	SUDESTE	25	75
	NORDESTE	7	93
	SUL	20	80
	NORTE	7	93
	CENTRO-OESTE	21	79
	Até R\$ 415	1	99
RENDA FAMILIAR	R\$ 416 a R\$ 830	4	96
	R\$ 831 a R\$ 1245	16	84
	R\$ 1.246 a R\$ 2.075	33	67
	R\$ 2.176 a R\$ 4.150	58	42
	R\$ 4.151 ou mais	81	19
	A	91	9
CLASSE SOCIAL	B	58	42
	C	16	84
	D-E	1	99

Fonte: TIC (2008)

Da Tabela 5 observa-se, além do um baixo percentual geral de acesso à Internet - menos de 20% dos domicílios - as disparidades envolvendo as faixas de renda familiar, a

zona urbana e rural e também entre as regiões geográficas, caracterizando uma distribuição concentrada dos recursos tecnológicos nas mãos de uma parcela restrita da população.

Neste sentido, entre as premissas estabelecidas pela norma N07 do SBTVD - abordada no Capítulo 4 - para os dispositivos receptores, duas delas vêm ao encontro desta realidade: atender aos diversos cenários existentes no país e acomodar o maior número possível de soluções tecnológicas. A intenção é promover a inclusão digital tanto das camadas mais pobres da população quanto das regiões de difícil acesso aos meios de comunicação, atendendo as exigências impostas pelo governo nos decretos que estabeleceram a implantação do SBTVD, que prevê o acesso livre, gratuito e para toda a população (BEDICKS JUNIOR, 2008; BRACKMANN, 2008; ETCHARTE, 2008; KUNG; KENZO; ZANINI, 2007; LAMAS et al, 2005).

Enquanto nas regiões metropolitanas a teledensidade é bastante alta, muitos municípios do interior do Brasil sequer possuem acesso à telefonia fixa, dependendo muitas vezes de tecnologias alternativas. Isto denota a heterogeneidade da malha de comunicação nacional e suas peculiaridades regionais, sendo inviável a eleição de um ou outro meio exclusivo ou oficial como canal de retorno, devendo ser levadas em consideração as características inerentes a cada região geográfica ou nível socioeconômico (INTERVOZES, 2006; LAMAS et al, 2005; MARGALHO; FRANCÊS; COSTA, 2007; MONTEZ; BECKER, 2005).

De acordo com a norma N07, os receptores devem ser compatíveis com toda a infraestrutura de telecomunicações brasileira e tecnologias envolvidas, em uma arquitetura híbrida envolvendo múltiplos meios de transmissão e provedores de acesso (BEDICKS JUNIOR, 2008; LAMAS et al, 2005), conforme ilustrado pela Figura 26, na próxima página. Entre os meios tecnológicos que podem ser utilizados, alguns em pleno funcionamento e

outros ainda em fase de pesquisas, os mais conhecidos são (BRACKMANN, 2008; MONTEZ; BECKER, 2005; SANTOS et al, 2009):

- a) telefonia fixa;
- b) ADSL;
- c) rádio;
- d) satélite;
- e) *Power Line Communication* (PLC);
- f) rede celular: GSM, *General Packet Radio Service* (GPRS), *Code Division Multiple Access* (CDMA) e outros;
- g) redes *Wireless Fidelity* (WiFi) e *Worldwide Interoperability for Microwave Access* (WiMAX);
- h) redes ad-hoc.

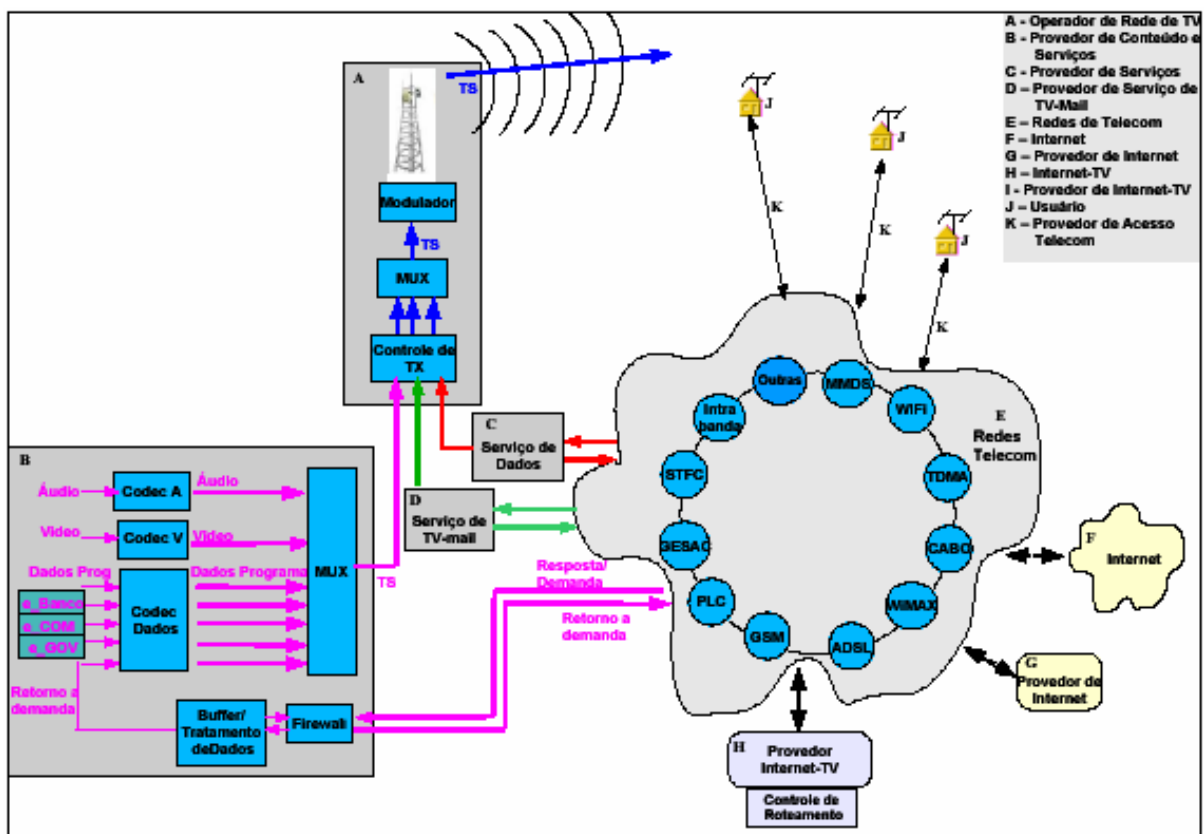


Figura 26. Arquitetura do canal de interatividade e as múltiplas tecnologias alternativas
Fonte: ETCHARTE, V. (2009, p. 05)

O emprego de uma ou outra tecnologia, além dos aspectos econômicos e geográficos discutidos, também pode variar de acordo com a modalidade de interatividade implementada, podendo ser intermitente, onde o usuário somente envia informações ao emissor, de forma assíncrona, unidirecional e não dedicada (votação de uma enquete, por exemplo) ou de interatividade permanente, exigindo um canal de retorno dedicado, em tempo integral e que proporcione comunicação bidirecional, permitindo mais poder ao usuário em virtude de possibilitar a solicitação de conteúdo personalizado (que será enviado pelo emissor via difusão ou pelo próprio canal de retorno) ou até mesmo de sua participação na programação, com o envio à emissora de conteúdos de sua autoria, via canal de retorno (ALVES, 2008; KUNG; KENZO; ZANINI, 2007).

A complexidade das aplicações envolvidas também deve ser levada em conta nesta questão. Logicamente, a adoção pelo usuário de um canal de retorno em banda larga proporcionará uma interatividade em grau mais alto, com mais alternativas e mídias de qualidade mais alta. Por sua vez, aplicações interativas mais simples, como votação, enquetes, escolha de ângulos de transmissão, acesso à Internet ou a aplicações governamentais, por exemplo, funcionam satisfatoriamente com bandas abaixo de 512 Kbps. Sendo assim, a demanda por conteúdos interativos tende a variar de acordo com as possibilidades financeiras do usuário (MONTEZ; BECKER, 2005).

No próximo capítulo a pesquisa desce um nível na escala de detalhamento, adentrando no Sistema Brasileiro de Televisão Digital, relatando o caminho percorrido no processo de digitalização da televisão brasileira e também as características e normas técnicas definidas para o padrão brasileiro ISDB-Tb.

4 SISTEMA BRASILEIRO DE TV DIGITAL E PADRÃO ISDB@TB

O Sistema Brasileiro de Televisão Digital, baseado no padrão japonês foi lançado oficialmente em 2006, pelo decreto presidencial N°. 5.820 (LOPES, 2007). Entretanto, um longo caminho tem sido percorrido desde os anos 90 até a transmissão inaugural em sistema aberto - ocorrida em 2 de dezembro de 2007 na cidade de São Paulo -, envolvendo o Governo Federal, instituições de pesquisa e membros da cadeia televisiva e da indústria eletrônica. A próxima seção aborda o processo evolutivo dos trabalhos em torno da definição de um padrão para ser implantado em nosso país.

4.1 ABORDAGEM HISTÓRICA

Nos anos 90, graças a pesquisas iniciadas nos anos 80 ou mesmo antes, os primeiros sistemas de TV em alta definição e digital nasceram, tendo como pioneiros os Estados Unidos, União Europeia e Japão, conforme visto no Capítulo 2. Diante deste quadro, e considerando a parcela populacional com acesso ao meio televisivo, iniciam-se no Brasil, nos anos 90, as pesquisas em prol da implantação da televisão em alta definição no país, por iniciativa do Governo Federal. O primeiro órgão instituído para tal finalidade foi a Comissão Assessora para Assuntos de Televisão (Com-Tv), pelo Ministério das Comunicações em 1991, para estudar os sistemas que estavam sendo desenvolvidos nos países pioneiros (BREUNIG; MARQUES, 2008; LOPES, 2007).

Em 1994, formou-se o Grupo Técnico Abert/Set de TV Digital, composta por profissionais das áreas de telecomunicações, televisão, rádio e multimídia, com o objetivo de analisar os sistemas de TV Digital que estavam sendo desenvolvidos, o seu processo de implantação nos países adotantes e traçar comparativos entre os sistemas estudados. O

resultado seria a adoção do mais adequado entre os três sistemas considerados, e consequente implantação no Brasil (BREUNIG; MARQUES, 2008; LOPES, 2007).

Em 1997 a Com-Tv foi extinta, e os trabalhos passaram a ser conduzidos pela Agência Nacional de Telecomunicações (ANATEL), ainda com a intenção de adotar um dos padrões já desenvolvidos e implanta-lo aqui. Em 1998, o grupo Abert/Set firmou um convênio com a Universidade Presbiteriana Mackenzie, de São Paulo, que passou a ser responsável a partir de 1999 por uma série de testes laboratoriais, pesquisas de campo e demonstrações públicas envolvendo os sistemas abordados. Nesta fase, o CPqD colaborou com a parte metodológica envolvendo a pesquisa, análise de resultados e regulamentação/definição dos aspectos técnicos a serem considerados nos testes (BREUNIG; MARQUES, 2008).

Foram realizados testes com os padrões envolvidos – americano, europeu e japonês – envolvendo avaliações e testes técnicos, demonstrações públicas em *shopping-centers* e pesquisas junto a usuários, fabricantes e empresas do ramo difusor. A aferição dos resultados levou em consideração os seguintes quesitos (BREUNIG; MARQUES, 2008):

- a) comportamento com interferência;
- b) robustez do sistema digital a interferências por multipercurso;
- c) peculiaridades de desempenho de recepção;
- d) características de desempenho e transmissão;
- e) traços do sistema/sinal para recepção móvel.

Após esta etapa, o sistema ATSC norte-americano foi praticamente descartado pela Anatel, devido à baixa imunidade a ruídos, pouca flexibilidade e recepção doméstica prejudicada. Por sua vez, os sistemas DVB europeu e ISDB japonês apresentaram melhor robustez de sinal, além de oferecer melhor suporte à interatividade e possibilidades de portabilidade e mobilidade (BARBARA; SOARES, 2008; BREUNIG; MARQUES, 2008).

Até então, as pesquisas vinham sendo conduzidas pelo governo e entidades envolvidas tendo como meta a adoção de um dos padrões de transmissão terrestre pré-existentes. Em 2003 o novo governo, levando em conta a intenção de disponibilizar a tecnologia para toda a população com custo baixo e atendendo a todos os aspectos inerentes à realidade nacional, após discussões sobre o tema envolvendo vários interessados, optou pelo desenvolvimento de um padrão nacional, publicando o Decreto Nº 4.901 de 26 de novembro do mesmo ano que instituiu oficialmente o Sistema Brasileiro de Televisão Digital – SBTVD (ARNOLDO, 2008; LOPES, 2007).

Os objetivos contemplados pelo decreto e que deveriam ser observados pelos grupos envolvidos na pesquisa foram:

“[...]”

I – Promover a inclusão social, a diversidade cultural do País e a língua pátria por meio do acesso à tecnologia digital, visando à democratização da informação;

II – Propiciar a criação de rede universal de educação à distância;

III – Estimular a pesquisa e o desenvolvimento e propiciar a expansão de tecnologias brasileiras e da indústria nacionais relacionadas à tecnologia de informação e comunicação;

IV – Planejar o processo de transição da televisão analógica para a digital, de modo a garantir a gradual adesão de usuários a custos compatíveis com sua renda;

V – Viabilizar a transição do sistema analógico para o digital, possibilitando às concessionárias do serviço de radiodifusão de sons e imagens, se necessário, o uso de faixa adicional de radiofrequência, observada a legislação específica;

VI – Estimular a evolução das atuais exploradoras de serviço de televisão analógica, bem assim o ingresso de novas empresas, propiciando a expansão do setor e possibilitando o desenvolvimento de inúmeros serviços decorrentes da tecnologia digital, conforme legislação específica;

VII – Estabelecer ações e modelos de negócios para a televisão digital adequados à realidade econômica e empresarial do País;

VIII – Aperfeiçoar o uso do espectro de radiofrequências;

IX – Contribuir para a convergência tecnológica e empresarial dos serviços de comunicações;

X – Aprimorar a qualidade de áudio, vídeo e serviços, consideradas as atuais condições do parque instalado de receptores no Brasil; e

XI – Incentivar a indústria regional e local na produção de instrumentos e serviços digitais [...]” (BRASIL, 2003).

Neste sentido, a responsabilidade pela coordenação dos trabalhos de pesquisa ficou a cargo do CPqD. Entretanto, tais trabalhos tiveram início somente no ano de 2005, dos quais tomaram parte 22 consórcios envolvendo 105 universidades e centros de pesquisa, além

de empresas nacionais e estrangeiras relacionadas à televisão e eletrônica, envolvidos no estudo do comportamento dos três padrões já existentes e definição de alternativas conceituais e tecnológicas adequadas ao contexto nacional (ALVES, 2008; CRUZ, 2009; IDEC, 2006).

O resultado destes estudos bem como de uma série de consultas junto a entidades representantes da cadeia televisiva, debates, reuniões e fóruns, serviram de base para o governo definir as características técnicas do padrão nacional a ser desenvolvido, levando em consideração as diretrizes propostas pelo decreto: acessibilidade por toda a população, inclusão digital, preservação da identidade nacional e fortalecimento da cadeia produtiva nacional (CRUZ, 2009; IDEC, 2006).

Devido a questões envolvendo atrasos no cronograma, pressões feitas pelas emissoras de TV e urgência em se definir um padrão, a ideia de desenvolver um padrão genuinamente nacional foi descartada, optando-se por um padrão baseado no ISDB japonês, acrescentando-se melhorias e inovações nacionais, de forma atender às necessidades expressas pelas premissas do Decreto 4.901 (LOPES, 2007).

O padrão japonês foi escolhido por oferecer maior robustez de transmissão, portabilidade e mobilidade e menor custo, além de permitir a incorporação das inovações propostas pelos grupos de pesquisa (LOPES, 2007). Sendo assim, em 29 de junho de 2006 o governo publicou o Decreto N. 5820, implantando oficialmente o Sistema Brasileiro de Televisão Digital Terrestre (SBTVD-T) e o padrão nipo-brasileiro ISDB@Tb – *Brazilian Integrated Services Digital Broadcasting Terrestrial* (ALVES, 2008; BRASIL, 2006; LOPES, 2007).

4.2 CARACTERÍSTICAS TÉCNICAS

O padrão ISDB@Tb consiste em um conjunto de especificações técnicas, cujo

objetivo é a definição de um sistema de transmissão que atenda os requisitos definidos e permita a recepção do sinal de forma abrangente por dispositivos dos mais variados tipos, sendo eles fixos, móveis ou portáteis (BEDICKS JUNIOR, 2008). A Figura 27 apresenta o diagrama de blocos do padrão brasileiro sob o ponto de vista dos conjuntos de especificações e seus inter-relacionamentos:

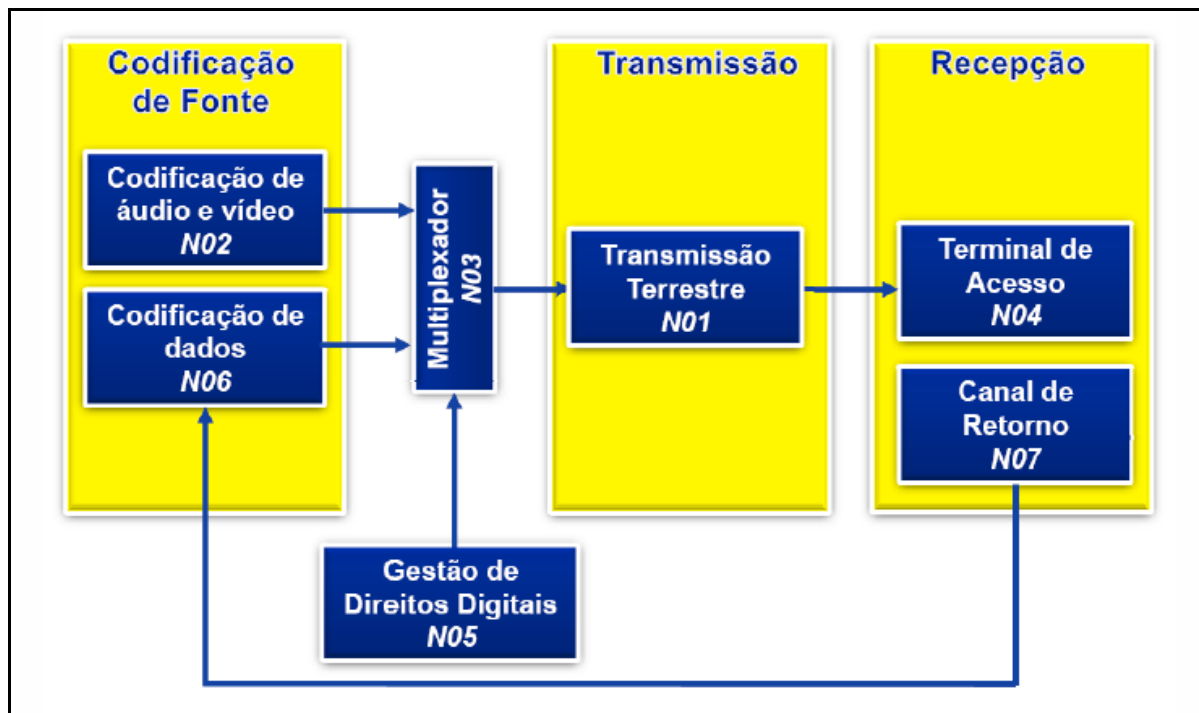


Figura 27. Diagrama de blocos das sete normas/especificações do ISDB-Tb
Fonte: BEDICKS JUNIOR, G. (2008, p. 61)

Essas especificações são frutos do trabalho dos Grupos de Trabalho do Módulo Técnico do Fórum SBTVD (instituído por ocasião do Decreto de 2003), e foram regulamentadas posteriormente sob a forma de Normas pela Associação Brasileira de Normas Técnicas (ABNT) (BEDICKS JUNIOR, 2008). Cada conjunto especifica características inerentes aos diversos aspectos tecnológicos envolvidos, resultando em sete especificações e respectivas normas ABNT, conforme discriminado pela Tabela 6:

Tabela 6. Descrição das especificações do ISDB-Tb e respectivas normas ABNT

ESPECIFICAÇÃO	ASPECTO	BASEADO NA NORMA	NORMA ABNT
N01	Transmissão terrestre	ARIB STD-B31	ABNT NBR 15.601
N02	Codificação de áudio e vídeo	ITU-T H.264; ISO/IEC 14496-3	ABNT NBR 15.602
N03	Multiplexação e serviços de informação	ARIB STD-B10	ABNT NBR 15.603
N04	Receptores	ARIB STD-B10; ITU-T H.264; ISO/IEC 14496-3	ABNT NBR 15.604
N05	Proteção de conteúdo – DRM	ARIB STD-B25	ABNT NBR 15.605
N06	Codificação de dados e especificação de transmissão	ARIB STD-B24; ETSI TS 101 812 (GEM); ETSI TS 102 819	ABNT NBR 15.606
N07	Canal de interatividade	ARIB STD-B21; ARIB STD-B24; ETSI TS 101	ABNT NBR 15.607

Fonte: Adaptado de BEDICKS JUNIOR, G. (2008, p. 61-62)

A especificação N1 estabelece como técnica de modulação a técnica *Band-Segmented Transmission Orthogonal Frequency Division Multiplexing* (BST-OFDM), padrão de codificação de transmissão terrestre derivado do OFDM/COFDM e desenvolvido no intuito de oferecer uma transmissão hierárquica, flexibilidade de serviços e maior imunidade a interferências (BEDICKS JUNIOR, 2008; BRACKMANN, 2008). Neste padrão, o canal de 6 MHz é dividido em 14 segmentos, sendo 13 reservados para transmissão e o segmento restante funcionando como banda de guarda ao redor do canal, conforme se pode observar na Figura 28. Esta segmentação possibilita a transmissão de até três serviços com característica de modulação e robustez distintas simultaneamente, caracterizando a transmissão hierárquica.

Na Figura 28 podemos observar um exemplo, onde nove segmentos foram alocados para transmissão em HDTV destinada a receptores fixos, três reservados para transmissão SDTV (receptores móveis) e o segmento central em LDTV (conhecido também como *one segment*), para recepção portátil, sendo cada serviço modulado em um padrão diferente (BEDICKS JUNIOR, 2008; BRACKMANN, 2008).

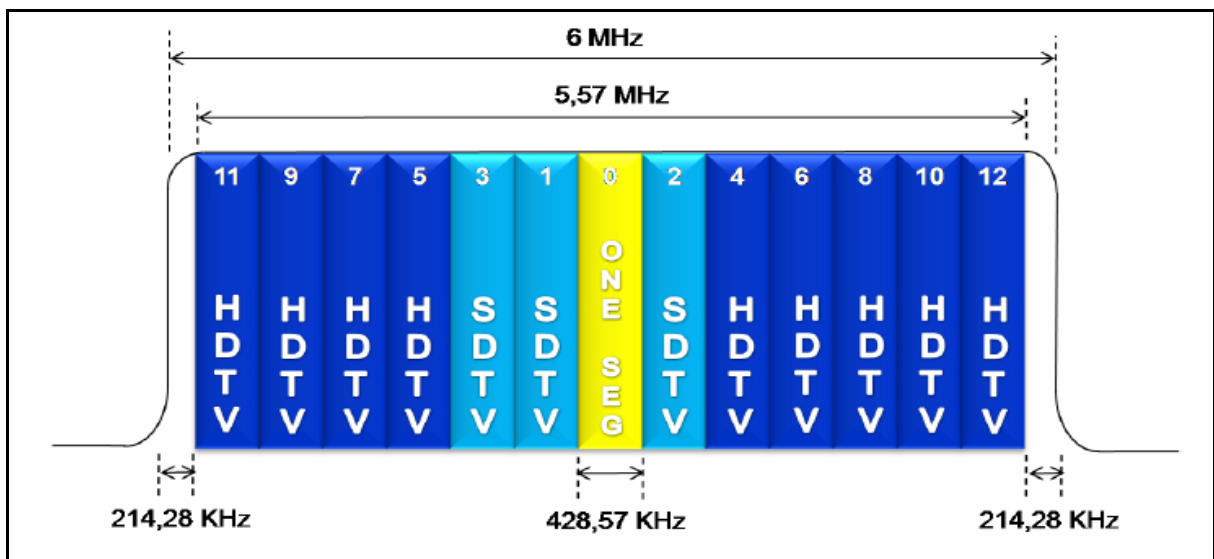


Figura 28. Transmissão OFDM com banda segmentada: BST-OFDM
 Fonte: BEDICKS JUNIOR, G. (2008, p. 62)

A especificação N2 estabelece os parâmetros relacionados à codificação de áudio e vídeo. Na questão do áudio, o padrão escolhido foi o MPEG-4 HE-AAC, considerado o sucessor do padrão MPEG-1 *Audio Layer 3* (MP3) e projetado para permitir uma reprodução de áudio em alta qualidade com menor taxa de bits (BEDICKS JUNIOR, 2008; BRACKMANN, 2008). No ISDB-Tb existem três possibilidades de taxas de áudio: 32 kHz, 44,1 kHz ou 48 kHz, com quantização mínima de sinal de entrada em 16 bits. Serviços de áudio adicional (como áudio em outro idioma e efeitos especiais) são possíveis, mas limitados ao número máximo de canais disponíveis do *bitstream* de áudio do qual pertencem (BEDICKS JUNIOR, 2008).

A especificação do padrão brasileiro prevê a utilização de dois perfis diferentes, de acordo com a capacidade do receptor, variando desde um até cinco canais simultâneos mais o canal de baixas frequências *subwoofer* – áudio 5.1, proporcionando mais realismo - conforme podemos observar na Tabela 7:

Tabela 7. Perfis do *codec* de áudio MPEG-4 HE-AAC

	RECEPTORES FIXOS E MÓVEIS	RECEPTORES PORTÁTEIS
PADRÃO	ISO/IEC 14496-3 (MPEG-4 AAC)	ISO/IEC 14496-3 (MPEG-4 AAC)
NÍVEL/PERFIL	AAC@L4 para multicanal 5.1 HE-AAC v1@L4 para estéreo	HE-AAC v2@L3 em dois canais
TAXA DE AMOSTRAGEM	48 kHz	48 kHz

Fonte: BRACKMANN, C. (2008, p. 30)

Para a camada de vídeo, o padrão adotado é o MPEG-4 H.264, considerado mais moderno e eficiente em relação à codificação MPEG-2 adotada pelos outros padrões de TV Digital (ZANCANARO; SANTOS; TODESCO, 2009). Embora tal *codec* seja padrão adotado para exibição de mídias de qualidade superior, como nos discos de *Blu-Ray* e *High-Definition/Density* DVD (HD DVD), ele foi desenvolvido de modo a permitir um escalonamento de transmissão em níveis diversos de qualidade e resolução, permitindo a variação da taxa de transmissão e níveis de resolução, possibilitando a recepção por dispositivos de capacidades variadas (BRACKMANN, 2008).

Neste sentido, dois perfis do padrão H.264 são empregados no ISDB-Tb: o perfil *High Profile* (HP), com taxa máxima de 25 Mbit/s (nível 4.0) e destinado à transmissão em alta definição e o perfil *Baseline Profile* (BP), destinado à transmissão para dispositivos móveis, portáteis e videoconferência via Internet, com nível de *bitrate* 1.3, correspondente à taxa máxima de 768 Kbit/s. (BRACKMANN, 2008).

Na Tabela 8 podemos observar a composição dos dois perfis e seus diferentes modos de resolução:

Tabela 8. Perfis do *codec* de vídeo MPEG-4 H.264

	RECEPTORES FIXOS E MÓVEIS	RECEPTORES PORTÁTEIS
PADRÃO	ITU-T H.264 (MPEG-4 AVC)	ITU-T H.264 (MPEG-4 AVC)
NÍVEL/PERFIL	HP@L4.0	BP@L1.3
RESOLUÇÃO E ASPECTO	480 (4:3 E 16:9); 720 (16:9); 1080 (16:9)	SQVGA (160x120 ou 160x90); QVGA (320x240 ou 320x180); CIF(352x288), todos em 4:3 e 16:9
TAXA DE QUADROS	30 e 60 Hz	15 e 30 Hz

Fonte: BRACKMANN, C. (2008, p. 29)

A especificação N3 define o MPEG-2 *System: Transport Stream* como padrão empregado na multiplexação dos fluxos, a exemplo dos sistemas antecessores que também empregam o referido padrão (BEDICKS JUNIOR, 2008; BRACKMANN, 2008). O processo de multiplexação já foi abordado anteriormente neste trabalho.

A especificação N4, por sua vez, define um conjunto de características técnicas necessárias aos receptores compatíveis com o ISDB-Tb, envolvendo aspectos eletrônicos, de *hardware*, *software* e arquitetônicos. Uma delas, por exemplo, diz respeito à quantidade mínima de memória volátil recomendada, que é de 2 MB para qualquer tipo de dispositivo. A lista é extensa, sendo algumas das características mandatórias e outras opcionais, variando de fabricante para fabricante ou conforme o tipo de dispositivo (BEDICKS JUNIOR, 2008).

A Figura 29 apresenta a configuração básica conforme a especificação N4.

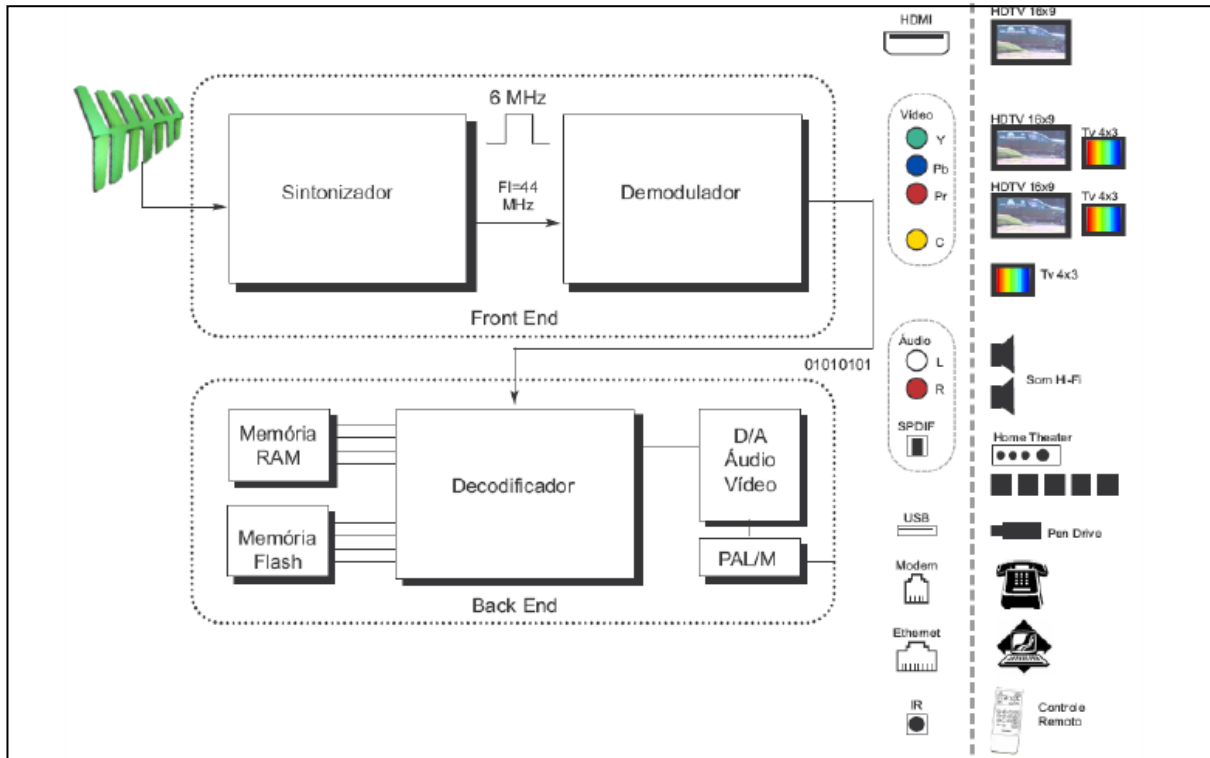


Figura 29. Configuração básica de um receptor ISDB-Tb
 Fonte: BEDICKS JUNIOR, G. (2008, p. 76)

A especificação N5 trata dos aspectos envolvendo direito de cópia de conteúdo. Foram definidos mecanismos de proteção em relação aos conteúdos transmitidos nas definições SDTV e HDTV. De acordo com a especificação, as interfaces de saída de áudio e vídeo digital dos receptores devem adotar protocolos de proteção de conteúdo específicos, de modo a impor barreiras contra a cópia e distribuição indiscriminada de conteúdos (pirataria). Para as interfaces de vídeo digital é obrigatório o uso do protocolo Intel *High-bandwidth Digital Content Protection* (HDCP). Já para o áudio digital, são mandatórios o HDCP e o protocolo *Serial Copy Management System* (SCMS) para o caso de interfaces *Sony-Philips Digital Interface Format* (SPDIF). Além das interfaces de áudio e vídeo, outras como *Internet Protocol* (IP), Firewire, e USB devem ser protegidas (BEDICKS JUNIOR, 2008).

A especificação N6 define o modelo de referência para a camada de *middleware*. O modelo adotado pelo padrão ISDB-Tb é o Ginga, desenvolvido no Brasil e baseado nas normas do conjunto N6 (BEDICKS JUNIOR, 2008) e que será apresentado no Capítulo 5.

A especificação N7 define os aspectos relacionados ao canal de interatividade. De acordo com a norma proposta, toda a infraestrutura de telecomunicações presente no país deve ser utilizada como aparato na transmissão de dados envolvendo o canal interativo, proporcionando a possibilidade de utilização de inúmeras tecnologias diferentes em virtude da variedade de sistemas de telecomunicação em operação no Brasil, abrangendo desde a telefonia convencional até redes sem fio WiMax, por exemplo. (BEDICKS JUNIOR, 2008).

Considerando esta flexibilidade necessária devido à heterogeneidade dos sistemas de telecomunicação, o conjunto N7 foi definido especificando as funcionalidades mínimas necessárias para o funcionamento do canal de interatividade. Estas normas devem ser atendidas pelos fabricantes de aparelhos receptores, de modo a permitir o funcionamento adequado do canal de interatividade independentemente do meio de comunicação empregado, ou pelo menos para o maior número possível de tecnologias ou meios de transmissão diferentes (BEDICKS JUNIOR, 2008).

A Figura 30 apresenta o modelo conceitual do ISDB-Tb para o ambiente do receptor, com suas camadas e tecnologias envolvidas:

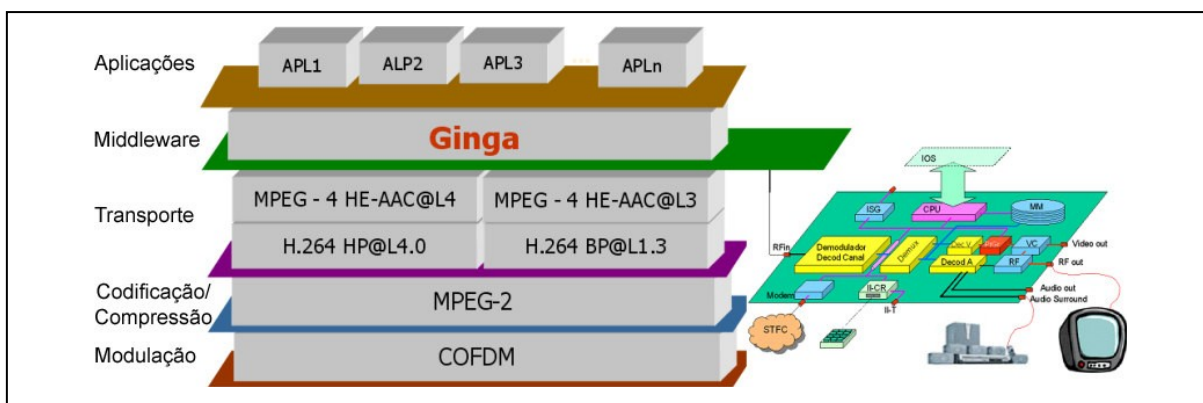


Figura 30. Camadas do aparelho receptor do SBTVD
Fonte: ARNOLDO, M. (2008, p. 04)

Compreendida a estrutura do ISDB-Tb, o próximo passo é abordar a sua camada de *middleware* – a inteligência do sistema – apresentada com detalhes no próximo capítulo.

5 O MIDDLEWARE GINGA

Conforme visto no Capítulo 2 a presença do componente de *middleware* é essencial no ambiente receptor, possibilitando a execução de aplicações (enviadas pelos estúdios via difusão) nos aparelhos receptores, abstraindo as funções das camadas mais baixas. Constitui-se como componente responsável pelo funcionamento do mecanismo de interatividade e é considerado como “a inteligência” por trás da TV Digital, permitindo além da execução de aplicações interativas a comunicação com servidores e provedores remotos, possibilitando uma oferta de serviços sob demanda e um *feedback* entre o usuário e a emissora, desde que haja um canal de retorno que suporte esta interatividade (MONTEIRO, 2009; ZANCANARO; SANTOS; TODESCO, 2009).

Cada padrão de TV Digital em funcionamento adota um *middleware* diferente. No caso do sistema Brasileiro, o componente adotado foi o Ginga[®], fruto de pesquisas coordenadas por duas frentes diferentes: o laboratório TeleMídia da Pontifícia Universidade Católica do Rio de Janeiro e o laboratório LAViD, da Universidade Federal da Paraíba (GINGA, 2008).

O nome foi escolhido em alusão às características inerentes ao povo brasileiro, como a forma do andar, falar e capacidade de se adaptar às situações diferentes, sendo considerada uma qualidade. Ginga também é um movimento característico da capoeira, uma mistura entre luta e dança de origem afro-brasileira, e a escolha do nome faz uma analogia entre a capoeira e a luta pela liberdade e igualdade do povo brasileiro, representando também o trabalho árduo realizado pelas equipes em busca de uma especificação genuinamente nacional, livre e que atenda às necessidades do ISDB-Tb, estabelecidas nas especificações ABNT N6 e N7 do SBTVD (BRACKMANN, 2008; GINGA, 2008; KNEIPP, 2009).

A especificação é constituída por um grupo de tecnologias padronizadas e de uma

série de inovações brasileiras, sendo considerado como a solução de *middleware* para TV Digital mais avançado da atualidade. A especificação é aberta e livre de *royalties*¹¹, adotando a licença *GNU General Public License, version 2 (GPLv2)*, permitindo a livre produção de conteúdo interativo para plataformas de *set-top boxes* heterogêneas de forma simplificada, impulsionando o processo de produção de aplicações interativas (BRACKMANN, 2008; GINGA, 2008).

O modelo de referência foi desenvolvido levando em conta a abrangência do meio televisivo junto à população brasileira, tido pelo governo como veículo oportuno no processo de inclusão social e digital em massa e de baixo custo, tendo em vista a carência de acesso a recursos tecnológicos enfrentados pela maioria da população brasileira, como acesso à Internet em banda larga. O objetivo seria fornecer à população acesso a serviços como educação à distância, informação e serviços sociais por meio da TV, sendo que algumas aplicações governamentais destinadas à inclusão digital, como o *T-Government* ou *Television Electronic Government (T-Gov)*, *Television Health (T-Health)* e *Television Learning (T-Learning)* já são suportadas pela especificação atual (BRACKMANN, 2008; GINGA, 2008; KNEIPP, 2009; MASCARENHAS; NORONHA; TAVARES, 2009).

Do ponto de vista arquitetônico, a estrutura é composta de três subsistemas: Ginga-NCL, máquina de apresentação que suporta o desenvolvimento e execução de aplicações de paradigma declarativo; Ginga-J, máquina de execução que dá suporte às aplicações de paradigma procedural e o subsistema *Common-Core (Ginga-CC)*, núcleo comum que lida com processos de nível inferior e provê serviços aos outros dois subsistemas, conforme ilustrado na Figura 31 exibida na próxima página (SOARES, 2008b).

Apesar da distinção entre ambientes de aplicações declarativas e procedurais, é possível o desenvolvimento de aplicações híbridas mesclando os dois paradigmas, graças ao

¹¹ Importância paga ao detentor ou proprietário pelo uso de um recurso, produto, marca ou patente (ROYALTIES, 2010).

emprego de um mecanismo lógico de ponte (*bridge*) estabelecido entre o Ginga-NCL e Ginga-J, permitindo o desenvolvimento de aplicações flexíveis. Deve-se considerar entretanto que o ambiente de execução procedural não é mandatória em dispositivos portáteis. (DUARTE, 2009; SOARES, 2008a; SOARES, 2008b). A Figura 31 apresenta a arquitetura de subsistemas do Ginga:

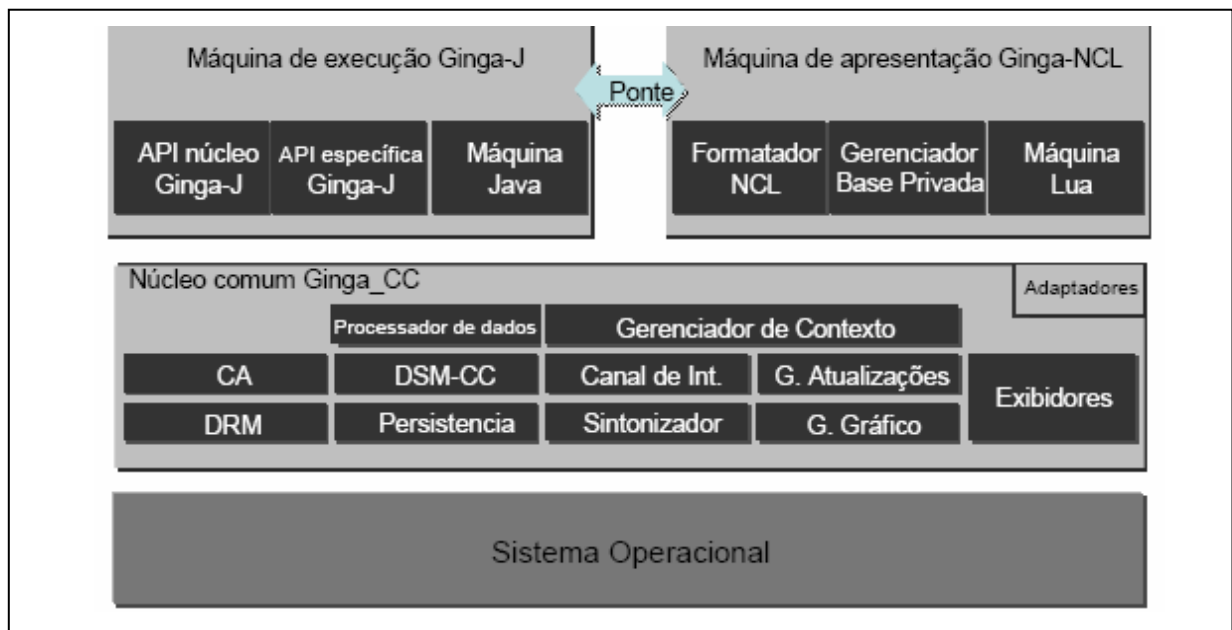


Figura 31. Arquitetura de referência do *middleware* Ginga
Fonte: SOARES, L. (2008b, p. 69)

5.1 SUBSISTEMA GINGA-CC

Common-Core é o subsistema posicionado na camada mais baixa do *middleware*, responsável por oferecer suporte básico aos outros dois subsistemas, abstraindo detalhes de baixo nível. Por intermédio dele é que os outros subsistemas obtêm acesso aos recursos de *hardware* da plataforma e também do conteúdo recebido (BARBARA; SOARES, 2008; BRACKMANN, 2008; DUARTE, 2009). Conforme se vê Figura 32, o subsistema é composto de 13 módulos, cada qual responsável por determinadas funções, listados a seguir (BARBARA; SOARES, 2008; SOARES, 2008b):

- a) sintonizador: responsável pela sintonização dos canais e escolha entre os fluxos de transporte transmitidos em um canal sintonizado;
- b) filtro de seções: suportam o acesso a partes específicas do conteúdo do fluxo de transporte, por solicitação das API's do Ginga para a execução de tarefas;
- c) processador de dados: responsável pelo acesso, processamento e repasse de dados recebidos pela camada física, bem como notificar os componentes do sistema sobre algum evento recebido que seja de interesse. Gerencia o processo de captura dos conteúdos transmitidos pela emissora via carrossel;
- d) persistência: capacidade de salvar conteúdo em formato de arquivos, disponibilizando o conteúdo para acesso em tempo oportuno;
- e) iniciador de aplicações: responsável pelo carregamento, configuração, inicialização, execução e controle do ciclo de vida de aplicações;
- f) adaptador de áudio/vídeo (A/V) principal: permite que as aplicações reconheçam o fluxo de áudio e vídeo, funcionalidade necessária para que as aplicações controlem suas ações de acordo com o fluxo de transmissão de A/V;
- g) gerenciador de gráfico: responsável pela definição de como as mídias – imagens, áudio e vídeo – serão apresentadas ao usuário;
- h) gerenciador de atualizações: responsável pela verificação das atualizações de versões disponíveis para o *middleware*, gerenciando o processo de *download* e instalação de atualizações e correções. Este módulo não é obrigatório;
- i) exibidores de mídia: ferramentas necessárias (*codecs*) para a exibição dos arquivos de mídia, como MPEG, *Joint Photographic Experts Group* (JPEG), *Graphics Interchange Format* (GIF), texto (TXT), HTML entre outros, incluindo a exibição dos fluxos MPEG-4 H.264;
- j) interface com o usuário: responsável pela captura e tratamento das interações e

comandos realizados pelo usuário – controle remoto, por exemplo - e transmitir aos demais módulos;

- k) gerenciamento de contexto: responsável pela captura das preferências do usuário, repassando aos outros subsistemas. Módulo também não obrigatório;
- l) canal de retorno: provê a interação entre os subsistemas de nível mais alto e o canal de retorno, bem como administrar o funcionamento deste canal;
- m) acesso condicional e direitos autorais: o primeiro fornece funcionalidade de bloqueio e restrição de conteúdos a serem exibidos. O segundo se refere aos protocolos de segurança e direitos autorais sobre cópia de conteúdo, já comentados anteriormente. O módulo de acesso condicional não é obrigatório.

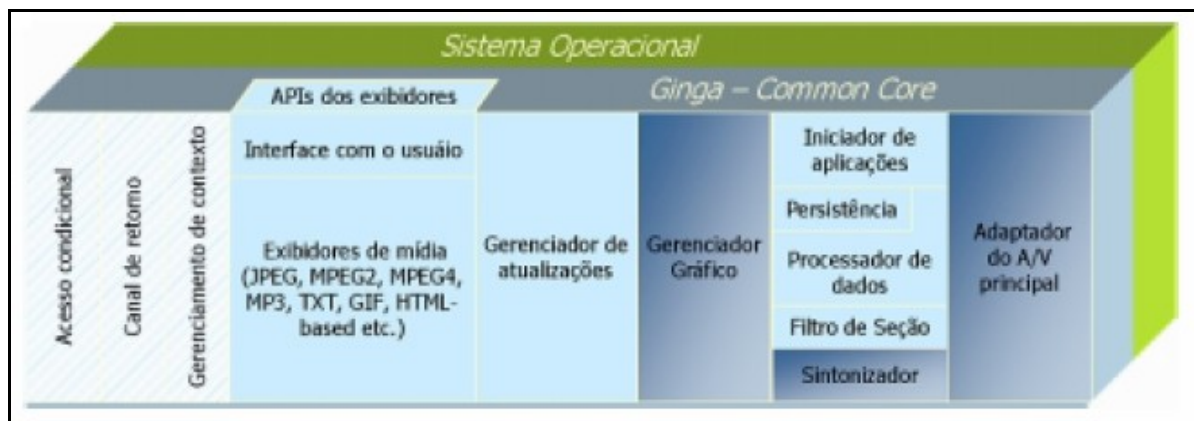


Figura 32. Arquitetura em camadas do *middleware* Ginga
 Fonte: BARBARA, F.;SOARES, L. (2008, p. 15)

5.2 SUBSISTEMA GINGA-NCL

O ambiente declarativo do Ginga, desenvolvido nos laboratórios da PUC-Rio, constituindo-se como a inovação totalmente brasileira do SBTVD. É responsável pelo suporte ao processamento de aplicações declarativas hipermídia escritas em linguagem NCL (*Nested Context Language*), baseada em XML e sua linguagem de *script* LUA, ambas desenvolvidas

pelo laboratório TeleMídia da PUC-Rio (MONTEIRO, 2009; SOARES, 2008b).

Sua característica principal é fornecer uma ligação e sincronismo temporal entre objetos de mídia que compõem um documento, em forma de módulos distintos e interligados, sendo considerados como objetos de mídia individual o fluxo de vídeo MPEG, o áudio, imagens JPEG e outros que compõem um documento, inclusive objetos escritos em linguagem imperativa, como aplicações Xlet, tornando-a adequada ao desenvolvimento de aplicações não lineares (CRUZ, 2009; MONTEIRO, 2009).

NCL é baseada no paradigma *Nested Context Model* (NCM) que utiliza o conceito de nós e elos para representar os documentos hipermídia, sendo os nós correspondentes aos objetos de mídia que compõem um documento e os elos, correspondentes aos relacionamentos de tempo/espço existentes entre estes nós (CARVALHO, 2008).

A linguagem oferece um nível de abstração mais alto comparado às linguagens procedurais, facilitando o trabalho de programação. Outros benefícios são: a facilidade de reuso devido à possibilidade de compartilhamento de objetos, geração e modificação dos documentos de forma instantânea (dispensando a compilação), maior separação entre o conteúdo e a estrutura de apresentação e suporte declarativo a múltiplos dispositivos de entrada e saída ou exibição (MONTEIRO, 2009; SOARES, 2008b). A arquitetura do subsistema Ginga-NCL pode ser vista na Figura 33:

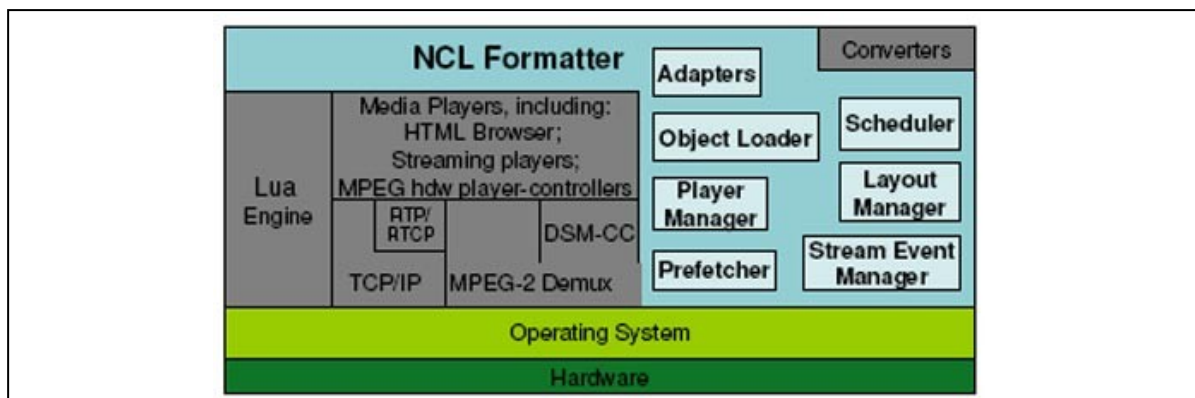


Figura 33. Arquitetura do Ginga NCL
Fonte: LIMA, F. (2007, p. 03)

O componente principal do subsistema é a máquina de interpretação declarativa, de responsabilidade do Formatador NCL (*NCL Formatter*), responsável pela exibição e controle das apresentações, mantendo o sincronismo entre os objetos de mídia que compõem o documento apresentado de acordo com o estipulado pelo programador. Outro componente importante é a máquina de execução Lua (*Lua engine*), responsável pelo processamento de código imperativo escrito nesta linguagem de *script*, considerada uma das mais eficientes e ideais em sua categoria, muito utilizada no desenvolvimento de jogos, além de possuir a API LuaJava, permitindo a manipulação de aplicações Java dentro do ambiente da linguagem Lua, colaborando com o processo de ponte estabelecido entre o subsistema declarativo e o imperativo. Por fim se tem o exibidor (*user agent*) XHTML, responsável pela interpretação de arquivos contendo HTML, *Cascading Style Sheets* (CSS) e *Javascript* (BARBARA; SOARES, 2008; CRUZ, 2009; DUARTE, 2009; GINGA-NCL, 2009; SOARES, 2008b).

O subsistema GINGA-NCL é mandatório para todos os receptores fixos, móveis e portáteis do SBTVD (CRUZ, 2009) e foi aprovado como padrão internacional para desenvolvimento de aplicações declarativas de televisão (TV) digital pela *International Telecommunication Union* (ITU/UIT) em maio de 2009 pela Recomendação H.761, ocasião considerada como marco na evolução tecnológica brasileira, abrindo portas para a exportação de tecnologia genuinamente nacional (CARVALHO, 2010; DUARTE, 2009).

5.3 O SUBSISTEMA GINGA-J

O GINGA-J é a porção do GINGA responsável pelo processamento de aplicações procedurais. Assim como nos demais *middlewares* para TV Digital, as aplicações são escritas em linguagem Java, utilizando APIs específicas para produção de aplicações executadas em aparelhos receptores. Estas aplicações são denominadas *Xlets*, e serão abordadas adiante, no

Capítulo 6 (ABNT, 2010; BECKER et al, 2005; KULESZA, 2009).

Java é a linguagem obrigatória dos ambientes procedurais em todos os padrões de TV Digital aberta do mundo, até então. Na realidade, Java não é apenas uma linguagem de programação, mas um conjunto de tecnologias que engloba uma linguagem de programação, coleções de APIs destinadas ao desenvolvimento de aplicações para diferentes plataformas (*desktop, mobile, web, tv* e outras) além de uma máquina de execução embarcada que possibilita rodar as aplicações Java *bytecode*, denominada Java *Virtual Machine* (JVM) (KULESZA, 2009). Java tem como principais características (KULESZA, 2009):

- a) sintaxe familiar, semelhante àquela da linguagem C;
- b) orientação a objetos;
- c) simplicidade e robustez;
- d) suporte nativo a *threads* e coleta de lixo;
- e) independência de plataforma, possibilitada pela compilação de um código intermediário denominado *bytecode*, interpretado pela JVM que serve como *middleware* entre a plataforma de *software*/sistema operacional e a aplicação, conforme ilustrado pela Figura 34, na página seguinte;
- f) sintaxe uniforme e fortemente tipada.

A Figura 34 ilustra as etapas do processo de geração de uma aplicação Java:

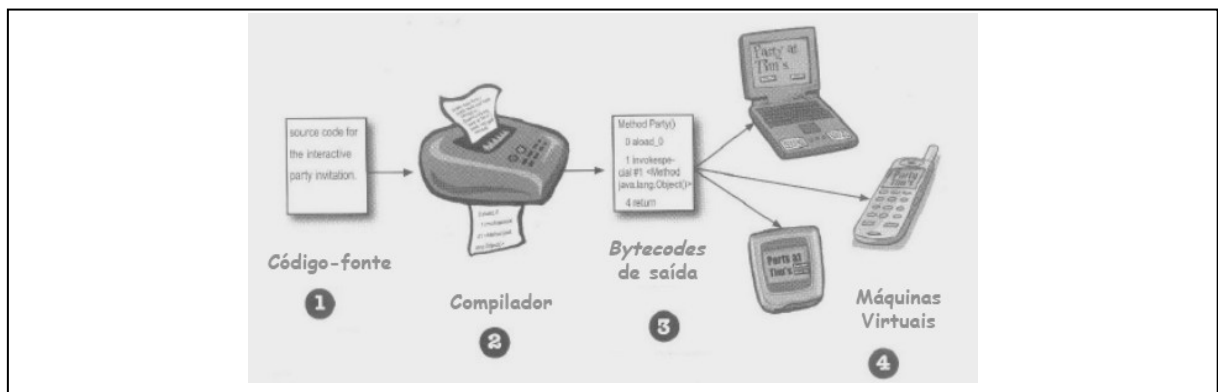


Figura 34. Etapas de uma aplicação Java
Fonte: KULESZA, R. (2009, p. 15)

O projeto Ginga-J foi concebido pelo laboratório LAViD, da UFBP, inicialmente sob o codinome de Flex-TV. Originariamente sua estrutura era baseada no *middleware* europeu (MHP), tendo como núcleo básico bibliotecas do *Globally Executable MHP* (GEM) - composto pelas APIs Java TV, *Home Audio Video* (HAVi), *Digital Audio Visual Council* (Davic) e extensões DVB -, mais as APIs *Java Media Framework* (JMF) 2.1, extensões GEM para suporte a fluxos de mídia, e os novos pacotes desenvolvidos pelos pesquisadores brasileiros (inovações nacionais). Além das APIs, uma máquina virtual Java completa o ambiente, funcionando como motor de processamento das aplicações escritas em Java (BOQUIMPANI, 2009; LIMA, 2007; ZANCANARO; SANTOS; TODESCO, 2009). O conjunto de APIs do Ginga-J foi subdividido em três grupos, conforme ilustra a Figura 35:

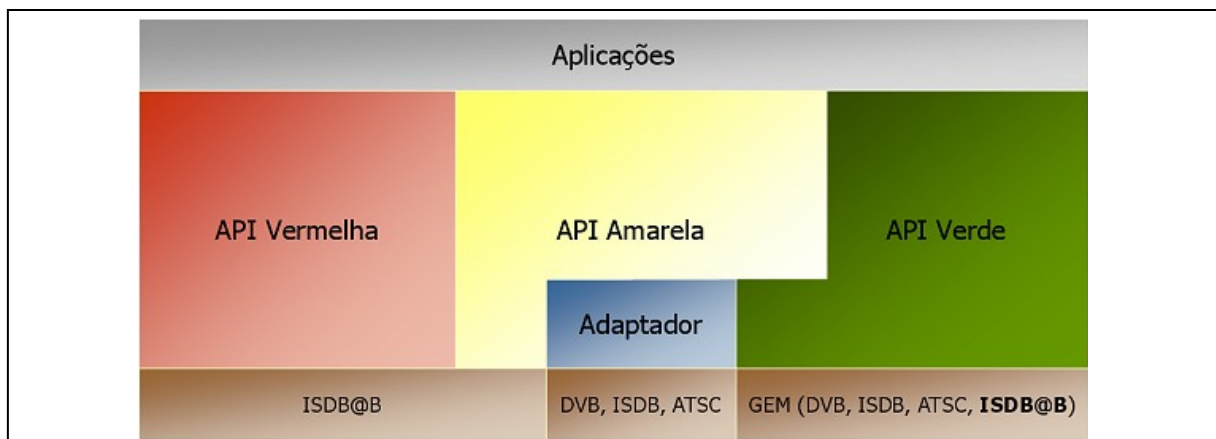


Figura 35. Conjuntos de API's que compõem o subsistema Ginga-J
Fonte: DUARTE, D. (2009, p. 35)

As APIs do conjunto verde, na constituição original, seguiam o núcleo comum do padrão GEM, a fim de proporcionar compatibilidade com os demais padrões de *middleware*. Dessa forma, aplicações escritas utilizando unicamente as APIs do conjunto verde funcionam normalmente em outros padrões de TV Digital aberta (LIMA, 2007; ZANCANARO; SANTOS; TODESCO, 2009). No projeto original, faziam parte as seguintes bibliotecas, provenientes do GEM:

a) Java TV: criada pela Sun Microsystems sobre a plataforma Java 2 *Micro Edition Platform* (J2ME), com a finalidade de prover acesso e funcionalidades aos receptores de TV Digital, com objetivo de desenvolver aplicações Xlet portáteis e independentes de tecnologia de difusão ou arquitetura de hardware (BRACKMANN, 2008; DARÓS, 2004; MONTEIRO, 2009; ZANCANARO; SANTOS; TODESCO, 2009).

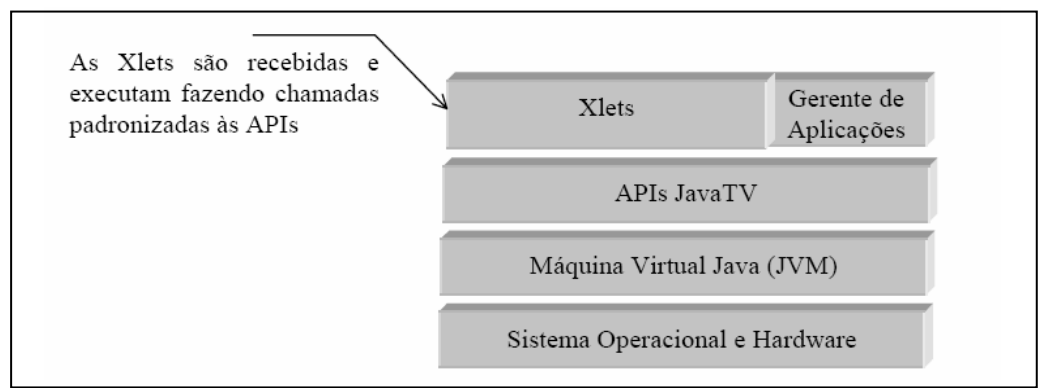


Figura 36. Xlets, gerente de aplicações e máquina virtual Java
 Fonte: BECKER, V. et al. (2005, p. 16)

Caracterizada por alto nível de abstração, pois não exige que o usuário lide com detalhes das camadas inferiores do ambiente, Java TV é composta por uma JVM, o gerente de aplicações do sistema e bibliotecas voltadas ao desenvolvimento de aplicações para o ambiente de receptores televisivos (ver Figura 36), oferecendo um conjunto de funcionalidades e facilidades ao desenvolvedor entre as quais podemos relacionar (BRACKMANN, 2008; DARÓS, 2004; MONTEIRO, 2009; ZANCANARO; SANTOS; TODESCO, 2009):

- tratamento de fluxo de áudio e vídeo;
- acesso condicional a dados do canal de transmissão;
- controle do sintonizador de canais;

- sincronização entre conteúdo audiovisual e aplicações;
- gerenciamento do ciclo de vida das aplicações, via gerente de aplicações do sistema;
- suporte a serviços e informações de serviços (onde um programa de televisão é considerado como um conjunto de componentes de mídia, denominado serviço);
- desenvolvimento de interfaces com o usuário com qualidade e alto nível de interatividade;
- controle de eventos do usuário;
- navegação pelo conteúdo do serviço.

- b) HAVi: *Home Audio Video*, desenvolvida por iniciativa das oito maiores companhias de produtos eletrônicos do mundo com o objetivo de padronizar a interconexão e interoperação de dispositivos de áudio e vídeo digital, permitindo que estes interajam entre si em uma espécie de rede doméstica – *Home Area Network* (HAN), onde os recursos e funcionalidades de cada integrante são compartilhados por todos, promovendo um controle facilitado para o usuário, possibilitando a este o acesso e controle de qualquer dispositivo integrado à rede via controle remoto ou outro dispositivo que interaja com o receptor/televisor. Também traz especificações relacionadas à apresentação de conteúdo e interface gráfica com o usuário. Para viabilizar o funcionamento do mecanismo de interação entre dispositivos heterogêneos, a especificação HAVi é independente de plataforma ou linguagem de programação (BRACKMANN, 2008; DARÓS, 2004; ZANCANARO; SANTOS; TODESCO, 2009);
- c) Davic: *Digital Audio Visual Council*, associação de desenvolvedores da indústria audiovisual, fundada em 1994 e extinta após cinco anos. O objetivo

foi a especificação de um padrão que proporcionasse interoperabilidade fim-a-fim de informações audiovisuais digitais, por difusão ou via canal de interatividade. A API Davic fornece acesso a funcionalidades do ambiente de TV Digital, como sintonizador de canais, acesso condicional, acesso ao fluxo de transporte e controle das seções DSM-CC. Outras funcionalidades importantes são: o gerenciamento de recursos escassos no ambiente receptor, especificação de formatos de conteúdo para objetos de áudio, vídeo, textos e hipertextos e facilidades no controle de língua adotada no áudio e legendas (BECKER et al, 2005; BRACKMANN, 2008; DARÓS, 2004; ZANCANARO; SANTOS; TODESCO, 2009);

- d) extensões DVB: API herdada do *middleware* europeu, visando suporte à segurança, acesso de dados e suporte a dispositivos de entrada ou saída (BRACKMANN, 2008).

As APIs amarelas, por sua vez, são bibliotecas incluídas durante o desenvolvimento do padrão brasileiro - algumas tendo como base bibliotecas do GEM (incluídas no grupo verde) e da especificação ARIB B.23 - mas compatíveis com outros sistemas, graças ao emprego de mecanismos de adaptação. Elas oferecem suportes a múltiplos usuários, dispositivos e redes, bem como ao recebimento e armazenagem de aplicações para serem executadas em momento oportuno. Também inclui o pacote JMF (*Java Media Framework*), que suporta o desenvolvimento de aplicações avançadas de áudio e vídeo, como captura de mídias e funcionalidades durante a apresentação, e também de modo parcial a especificação ARIB B.23, oriunda do *middleware* japonês, para fornecer funcionalidades relacionadas às tabelas de serviços MPEG contidos no *transport stream* (SOARES, 2008; ZANCANARO; SANTOS; TODESCO, 2009).

O conjunto vermelho contém as bibliotecas voltadas ao suporte das necessidades

específicas do contexto brasileiro, principalmente no que diz respeito à inclusão digital e social. Neste conjunto também foram incluídas as APIs que suportam o mecanismo de ponte entre as máquinas declarativa e procedural do Ginga, permitindo a interação entre aplicações Ginga e documentos NCL, em uma relação onde a aplicação Java pode agir como entidade inicial, podendo criar, modificar e construir documentos declarativos, ou agindo como entidade filha de um documento NCL, passando a atuar como objeto de mídia controlado pela aplicação declarativa, esta podendo inclusive fazer chamadas diretas aos métodos definidos no Xlet. As funcionalidades definidas pela API vermelha são suportadas somente pelo *middleware* Ginga, não podendo ser exportadas para os demais padrões de TV Digital aberta (ARNOLDO, 2008; DAMASCENO, 2008; SOARES, 2008).

Entretanto, em virtude da exigência do pagamento de *royalties* pelo uso das APIs definidas no núcleo GEM e conseqüentemente o aumento de custo de produção, situação que contraria as premissas estipuladas pelo governo bem como os objetivos do Fórum SBTVD, foi assinando um memorando de cooperação junto à Sun Microsystems em janeiro de 2008, solicitando a criação de uma especificação que substituísse o GEM e que fosse livre de *royalties*. Em junho do mesmo ano, formou-se a comissão Ginga-J e foi assinado o contrato de colaboração com a Sun, que se comprometeu em desenvolver a especificação Java DTV em parceria com o fórum, a fim de substituir as funcionalidades providas pelo GEM. A especificação foi entregue em dezembro de 2008, em tempo recorde, e ficou acordado que os direitos sobre a mesma seriam compartilhados entre a Sun e o Fórum SBTVD, além do uso ser liberado para implementações da especificação (BOQUIMPANI, 2009; INTEL, 2009; JAVA DTV, 2009; ZANCANARO; SANTOS; TODESCO, 2009).

Tal especificação é considerada de alto potencial de aceitação internacional, facilitando a convergência de tecnologias em torno da TV Digital, proporcionando maior liberdade de desenvolvimento. Para o Brasil, constituiu-se em um momento histórico,

colocando o país como liderança e potencial exportador de tecnologia na área, além do fato de ter desencadeado a criação da primeira especificação livre de *royalties* nesta categoria (BOQUIMPANI, 2009; INTEL, 2009; JAVA DTV, 2009; ZANCANARO; SANTOS; TODESCO, 2009).

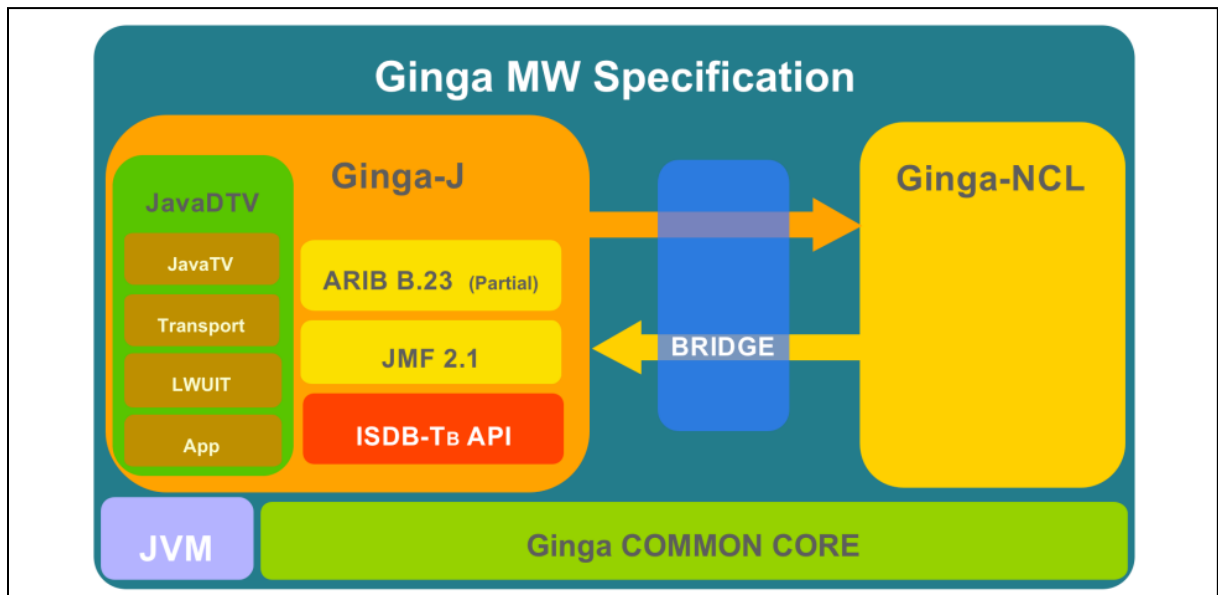


Figura 37. Estrutura do Ginga, após a substituição dos pacotes GEM por JavaDTV
 Fonte: ZANCANARO, A; SANTOS, P; TODESCO, J. (2009, p. 1093)

O Ginga-J passou por um período de consulta pública coordenada pela ABNT, entre junho e julho de 2009, recebendo uma votação de aprovação expressiva, restando finalizar alguns ajustes. A versão mais recente da especificação foi definida pela norma ABNT 15606-4, foi lançada em abril de 2010, em sua primeira edição válida a partir de 13 de maio de 2010 (ABNT, 2010; BOQUIMPANI, 2009), com uma estrutura reformulada e trazendo como núcleo principal as bibliotecas Java DTV (KULESZA; FERREIRA, 2009) conforme ilustrado na Figura 37.

O conjunto verde passou a ser constituído integralmente pelo conjunto Java DTV, englobando além de pacotes próprios, outros provenientes das especificações Java *Connected Device Configuration framework* (CDC) 1.1, FP 1.1 e PBP 1.1), Java TV 1.1, *Lightweight*

User Interface Toolkit (LWUIT), *Java Security Socket Extension 1.02*, *Java Cryptography Extension 1.0* e *Security and Trust Services for J2ME 1.0*, substituindo as bibliotecas GEM e oferecendo diversas funcionalidades, entre as quais (BOQUIMPANI, 2009; KULESZA; FERREIRA, 2009):

- a) interfaces baseadas em componentes;
- b) acesso a recursos de mais baixo nível da plataforma;
- c) catálogo de aplicações;
- d) controles de mídias;
- e) transação segura de dados;
- f) animações e transições gráficas ricas, devido à incorporação LWUIT (*LightWeight User Interface Toolkit*), desenvolvida pela Sun a princípio para implementar interfaces gráficas em dispositivos móveis. Foram retiradas algumas funcionalidades que não se encaixavam no ambiente televisivo, e incluídas inovações necessárias, substituindo o pacote HAVi;
- g) internacionalização das aplicações;
- h) suporte a manipulação de *smart cards*;
- i) fácil adaptação de conteúdo a outras tecnologias;
- j) interação simultânea de vários usuários;
- k) integração do receptor com outros dispositivos.

No grupo de APIs amarelas, permaneceram os conjuntos de pacotes brasileiros *br.org.sbtvd.net* e *br.org.sbtvd.net.si* baseados em ARIB 2.3, constituindo a chamada “API Informação de Serviço Dependente de Protocolo” juntamente com os pacotes ARIB B.23 dos quais esta API é dependente (ABNT, 2010). No grupo amarelo também está incluso o pacote *Java Media Framework 2.1*, conforme se observa na Figura 37.

No grupo vermelho, foi definida a permanência das seguintes bibliotecas (ABNT, 2010):

- a) *br.org.sbtvd.net.tuning*: especificação que define funcionalidades de sintonização de canais e utilização de componente *zapper* (navegador) para seleção e navegação entre os canais recebidos pelo receptor, além de informações relativas ao canal selecionado;
- b) *br.org.sbtvd.bridge*: contém classes e interfaces que dão suporte ao mecanismo de ponte entre as máquinas declarativa e procedural (*bridge*);
- c) *br.org.sbtvd.ui*: API de gerenciamento e suporte dos planos gráficos disponíveis no aparelho receptor. Dos quatro planos sobrepostos possíveis ilustrados na Figura 38, as aplicações Xlet do Ginga-J podem utilizar somente as três mais profundas, não podendo se sobrepor aos conteúdos da primeira camada (camada de legendas).

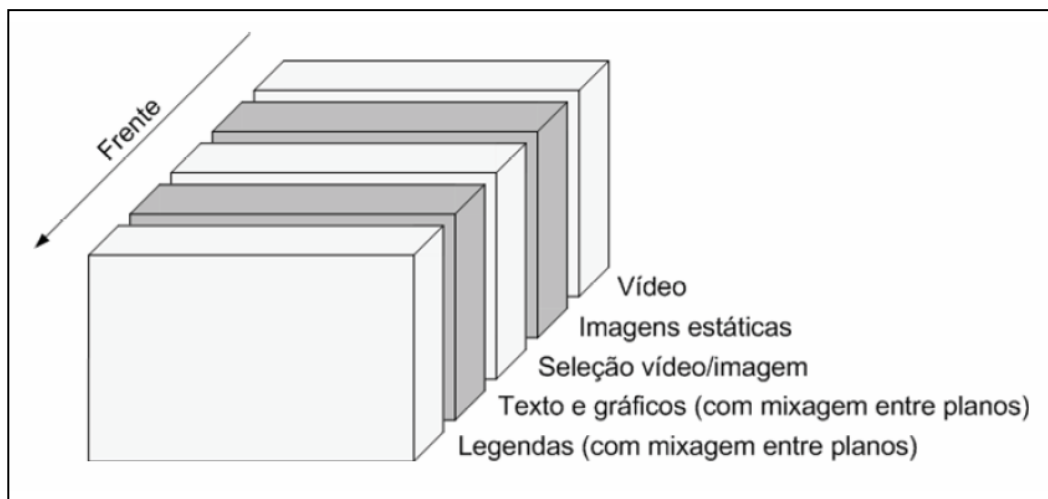


Figura 38. Estrutura de camadas para apresentação de serviços na tela do receptor
Fonte: ABNT (2010, p. 17)

A Figura 39 apresenta a estrutura de pacotes Ginga-J:

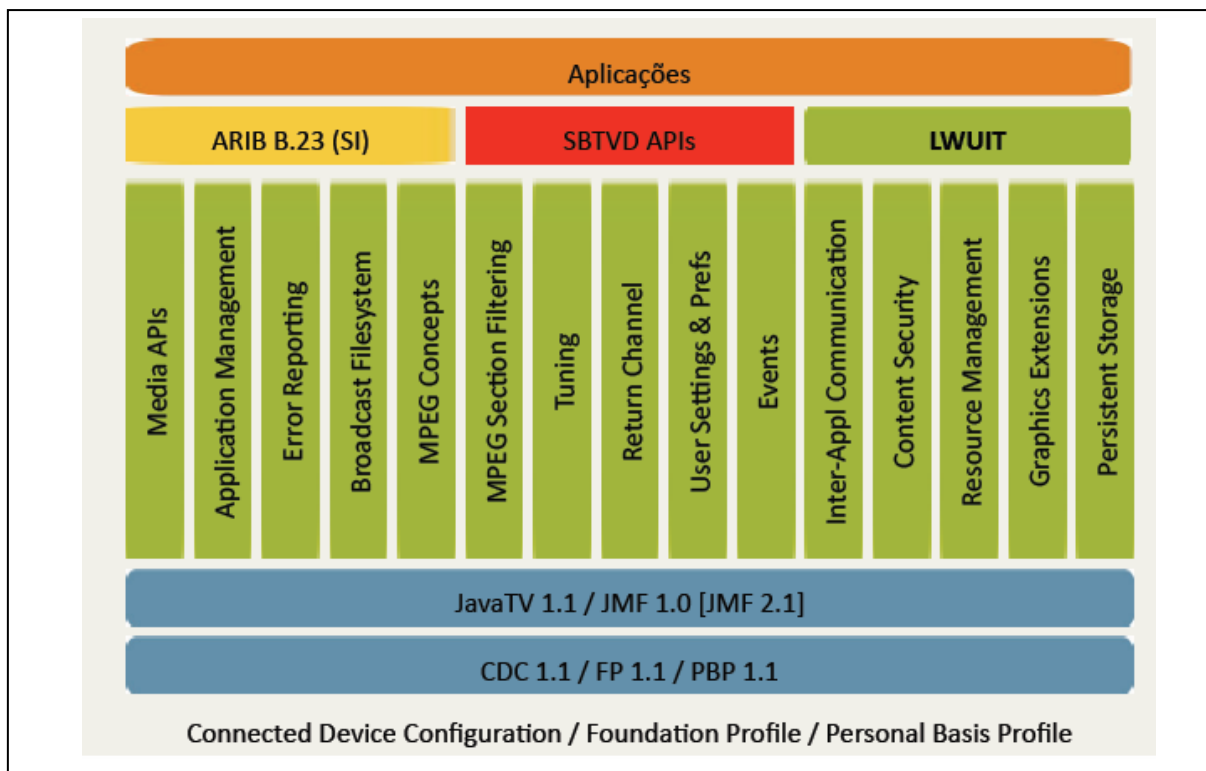


Figura 39. Estrutura de pacotes do *middleware* Ginga
 Fonte: KULESZA, R.; FERREIRA, J. (2009, p. 07)

Entre as funcionalidades oferecidas pelo ambiente Ginga-J, merece destaque o suporte ao desenvolvimento de aplicações avançadas, explorando a integração entre vários dispositivos. Nesse contexto, dispositivos como PDA, telefone celular e quaisquer outros que suportem Java podem ser integrados ao receptor/*set-top-box* via protocolos como *Bluetooth*, Wi-Fi, USB e outros (desde que suportados pelos dispositivos envolvidos) funcionando como uma rede doméstica (*home area network*), conforme esquematizando na Figura 40 da página seguinte. Para isto, é necessária a instalação de componentes Ginga no dispositivo periférico, para que esse possa ser devidamente cadastrado no receptor primário (que funcionará como elemento centralizador da rede), podendo a partir do momento em que for identificado, interagir com este, exercendo funções de controle remoto, ou até mesmo solicitando ao receptor fluxos de mídia para serem exibidos em sua interface (assistir TV no PDA, por exemplo) ou no caminho inverso, enviando ao centralizador fluxos de áudio ou vídeo capturados, conforme ilustrado na Figura 40 (NERY E SILVA et al, 2007).

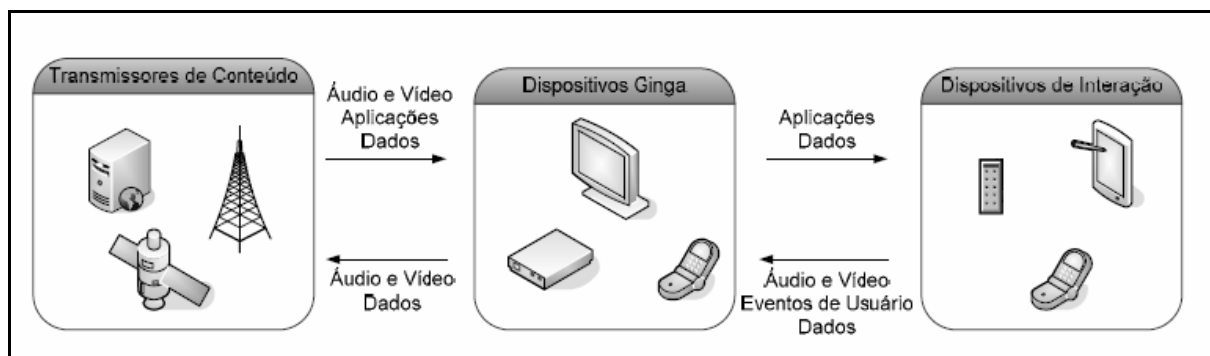


Figura 40. Contexto de arquiteturas envolvidas no modelo de execução do GINGA-J
 Fonte: NERY E SILVA, L. et al. (2007, p. 76)

Novamente, ressalta-se que para que o dispositivo possa ser integrado à rede, devem suportar a instalação de um componente GINGA que implemente o protocolo de comunicação entre dispositivos, para assim poder se registrar no dispositivo centralizador (receptor de TV ou *set-top box*) (NERY E SILVA et al, 2007).

5.4 MODELO DE APLICAÇÃO E AMBIENTE DE EXECUÇÃO GINGA-J

De acordo com o modelo de aplicação definido em Java TV e Java DTV e consequentemente pelo *middleware*, as aplicações procedurais são executadas em um ambiente orientado a serviços multitarefa, controladas por um componente denominado gerente de aplicações do sistema. Este modelo permite a execução simultânea de várias aplicações Xlet em uma mesma JVM, como se cada um estivesse rodando em sua própria máquina virtual. Assim, as aplicações simultâneas podem trabalhar de forma independente, competindo entre si pelos recursos comuns, ou trabalhar de maneira colaborativa, onde podem comunicar-se umas com as outras em busca de um mesmo resultado (ABNT, 2010; KULESZA, 2009).

A Figura 41 ilustra a arquitetura GINGA-J e sua relação com a plataforma:

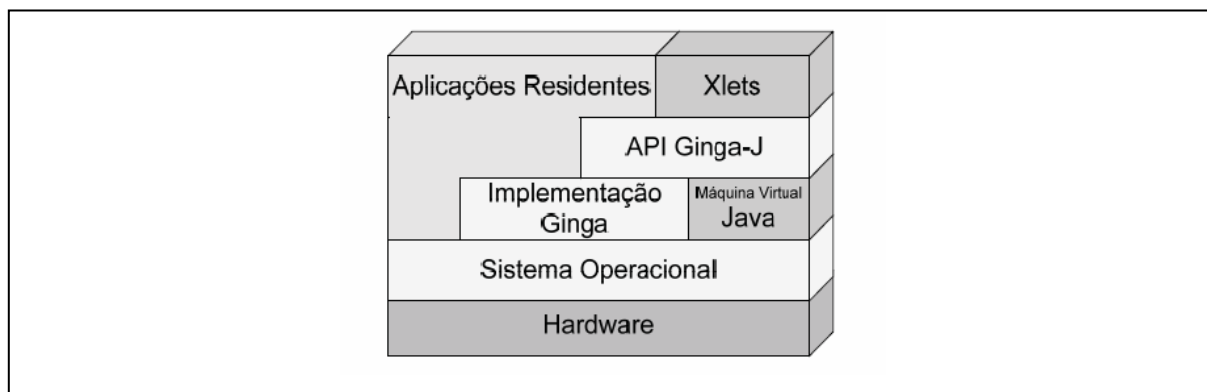


Figura 41. Arquitetura Ginga-J e ambiente de execução
 Fonte: ABNT (2010, p. 05)

O controle do ciclo de vida dos Xlets, que será detalhado na sequência, também é tarefa do gerente de aplicações, que deve receber e tratar as notificações relativas a cada aplicação. Estas podem ser controladas tanto pela emissora via canal de difusão, pelo mecanismo de *zapping* (navegação) e outros eventos gerados pelo usuário do sistema, como por outra aplicação Xlet e até mesmo por documentos declarativos NCL via mecanismo de ponte. Logo, a aplicação não tem controle sobre seu próprio ciclo de vida. Caso uma instância de determinada aplicação já esteja em execução no sistema, uma nova instância desta não pode ser inicializada, permanecendo inalterada aquela já em execução. Logo, todas as aplicações em execução recebem uma identificação e são registradas em um *namespace* definido por *javax.microedition.xlet.ixc.IxcRegistry*, atuando como um registro de nomes no domínio do sistema. Esse registro é essencial ao funcionamento do mecanismo de comunicação entre os Xlets, quando necessário à interação entre duas ou mais aplicações (ABNT, 2010; BECKER et al, 2005; DARÓS, 2004; KULESZA, 2009).

Uma aplicação Xlet deve estar sempre pronta para ser interrompida ou destruída. Em caso de interrupção, deve descartar os eventos gerados pelo usuário e sair da tela, reduzindo ao máximo seu consumo de recursos do sistema. Quando for destruída, todos os threads relacionados a ela devem ser interrompidos e os eventos assíncronos gerados em

relação a ela são cancelados. A aplicação é removida da tela e todos os recursos alocados por ela são liberados ao sistema (ABNT, 2010; KULESZA, 2009).

Cada aplicação deve possuir um contexto associado a ela, provendo assim um mecanismo pelo qual a aplicação possa obter informações sobre o ambiente onde está sendo executada bem como comunicar mudanças de estado para este ambiente. O contexto da aplicação é definido no momento da inicialização desta, quando uma instância da classe *javax.microedition.xlet.XletContext* é passada a ela pelo gerenciador de aplicações do sistema. Esta instância contém informações sobre o contexto e métodos.

As aplicações normalmente são recebidas pelo meio de difusão ou solicitadas via canal de retorno, sinalizadas para o terminal por intermédio da tabela denominada *Application Information Table* (AIT). Esta contém informações necessárias ao controle dinâmico do sistema sobre esta aplicação, além de informações adicionais de execução. Cada AIT por sua vez é inserida na tabela de mapeamento de programas (PMT), um catálogo dos serviços disponíveis no sistema - em execução ou de prontidão (ABNT, 2010; KULESZA, 2009).

O modelo de aplicações definido pela especificação GINGA-J permite que aplicações sejam armazenadas e executadas posteriormente. Aplicações também podem ser submetidas à *caching* pró-ativo, quando possível, com o objetivo de melhorar a velocidade de carga da aplicação no momento em que for solicitada, evitando latências (a aplicação demora a começar a rodar, como acontece na Internet de banda baixa) relacionadas ao tempo necessário para descarga (*download*) (ABNT, 2010).

As aplicações, quando residentes ou transmitidas pelo canal de retorno, devem ser devidamente autenticadas, certificadas e empacotadas em arquivos Java *ARchive* (JAR), sendo optativa a opção de compressão. Entretanto, quando forem transmitidas pela emissora via carrossel de objetos, não podem ser empacotadas em arquivos JAR, devendo ser mantida, porém, a organização original do pacote de arquivos (ABNT, 2010).

5.5 APLICAÇÕES XLET

As aplicações desenvolvidas para TV Digital no ambiente procedural são denominadas Xlets, de construção, comportamento e ciclo de vida semelhantes à Applets Web e MIDlets dos dispositivos móveis. Normalmente uma aplicação Xlet consiste de uma classe que implementa a interface *javax.tv.xlet.Xlet*, da biblioteca Java TV (ABNT, 2010; ABNT, 2007; BECKER et al, 2005; KULESZA, 2009).

No SBTVD, entretanto, alguns pacotes de Java TV tornaram-se obsoletos, entre eles *javax.tv.xlet* quando da liberação de Java DTV pela Sun a pedido do Fórum SBTVD. Sendo assim, as aplicações desenvolvidas para o SBTVD, de acordo com a Norma Brasileira (NBR) 15606-4 válida a partir de 13 de maio de 2010, devem implementar a interface *javax.microedition.xlet* ao invés de *javax.tv.xlet.Xlet* (ABNT, 2010; ABNT, 2007; BECKER et al, 2005; KULESZA, 2009).

Considerando a implementação via interface *javax.microedition.xlet*, a classe da aplicação deverá conter ao menos quatro métodos, que provêm o ciclo de vida da aplicação e que já são implementados pela interface, devendo ser sobrescritos na classe criada:

- a) *initXlet*;
- b) *startXlet* ;
- c) *pauseXlet* ;
- d) *destroyXlet*.

Por meio destes métodos, as aplicações Xlet poderão ser controladas pelo gerente de aplicações global do sistema, pelo *zapper* (geralmente o controle remoto), pela emissora que o transmitiu e até mesmo por outra aplicação Xlet ou documento declarativo do tipo NCL (ABNT, 2010).

Este controle sobre o ciclo de vida da aplicação é ilustrado na Figura 42:

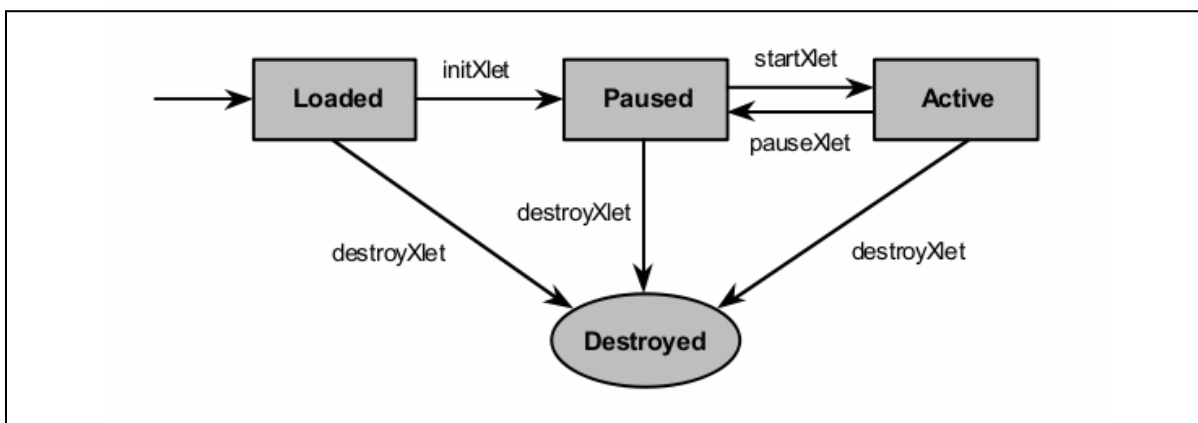


Figura 42. Diagrama com estados do ciclo de vida de um Xlet
 Fonte: ABNT (2010, p. 06)

O ciclo inicia pela instanciação da classe pelo gerente de aplicação do sistema, durante a chamada do método construtor da classe. Neste instante, nenhuma operação ou inicialização de recursos deve ser feita na classe. Caso não ocorra uma exceção, a instância de Xlet passa para o estado *Loaded*, caso contrário, é marcada como em estado *Destroyed* e descartada. Estando carregada (*Loaded*), a inicialização dos recursos necessários à aplicação bem como dos atributos da classe é feita chamando-se o método *initXlet*. Neste instante, o gerente de aplicações do sistema passa para a aplicação um objeto instância da classe *javax.microedition.xlet.XletContext*, definindo o contexto de execução desta dentro do ambiente da plataforma, servindo como um canal de comunicação entre a aplicação e o gerente de aplicações (ABNT, 2010; BECKER et al, 2005; DARÓS, 2004; PINTO; QUEIROZ-NETO; SILVA, 2007).

Após a inicialização, o gerente de aplicações passa o Xlet para o estado *Paused*, pronto para ser executado. A aplicação pode então ser executada, via chamada do método *startXlet*, passando para o estado *Active*, quando então a aplicação passa a funcionar, consumindo os recursos necessários à sua execução e sendo exibido no dispositivo, ou então ser encerrada, chamando-se o método *destroyXlet*. Estando ativo, o Xlet pode ser interrompido e voltar ao estado *Paused* invocando-se o método *pauseXlet*, ou destruído via

método *destroyXlet*, passando ao estado *Destroyed*, quando todos os seus recursos são liberados e a execução é finalizada (ABNT, 2010; BECKER et al, 2005; DARÓS, 2004; PINTO; QUEIROZ-NETO; SILVA, 2007).

5.6 A IMPLEMENTAÇÃO DE REFERÊNCIA OPENGINGA

OpenGinga é o nome dado à plataforma desenvolvida para suportar a execução de aplicações Ginga em computadores pessoais (PC), emulando o funcionamento de um ambiente *set-top box*, servindo assim como uma especificação de referência do Ginga (a princípio somente a máquina procedural) para ser utilizada pelos desenvolvedores como uma base para o desenvolvimento de aplicações Ginga (DE LUCA, 2009b; KULESZA; FERREIRA, 2009; OPENGINGA, 2010).

A especificação é baseada em projetos antecessores desenvolvidos pela equipe do laboratório LAViD da Universidade Federal da Paraíba, sendo conduzido atualmente pela rede de desenvolvedores do projeto Ginga CDN, submetido e aprovado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CETIC) em fevereiro de 2009, que a princípio trabalhava em conjunto com a empresa de software MOPA *Embedded Systems*, formada por ex-alunos da Universidade Federal da Paraíba (DE LUCA, 2009; KULESZA; FERREIRA, 2009; OPENGINGA, 2010). Ginga CDN é coordenado pela equipe do LAViD, objetivando a formação de um ambiente colaborativo via uma rede aberta de desenvolvedores coordenados na tarefa de implementar as funcionalidades do subsistema procedural do *middleware* Ginga (GINGACDN, 2009).

O OpenGinga foi desenvolvido para rodar em computadores com o sistema operacional Linux instalado, sendo que a implementação foi testada nas distribuições Debian, Fedora e Ubuntu. Na primeira versão liberada, a implementação era baseada nas bibliotecas

do GEM, sendo que a partir da liberação de Java DTV pela Sun, as bibliotecas GEM foram retiradas, e a equipe de desenvolvimento do Ginga CDN vem gradativamente adaptando as novas bibliotecas e disponibilizando a cada *release*. A máquina virtual utilizada pelo OpenGinga é a PhoneME, disponibilizada pela Sun na plataforma J2ME para rodar em dispositivos embarcados (DE LUCA, 2009; ENTENDENDO, 2008; OPENGINGA LIBERADO, 2009).

A arquitetura da primeira implementação de referência lançada após a substituição do GEM por Java DTV pode ser vista na Figura 43:

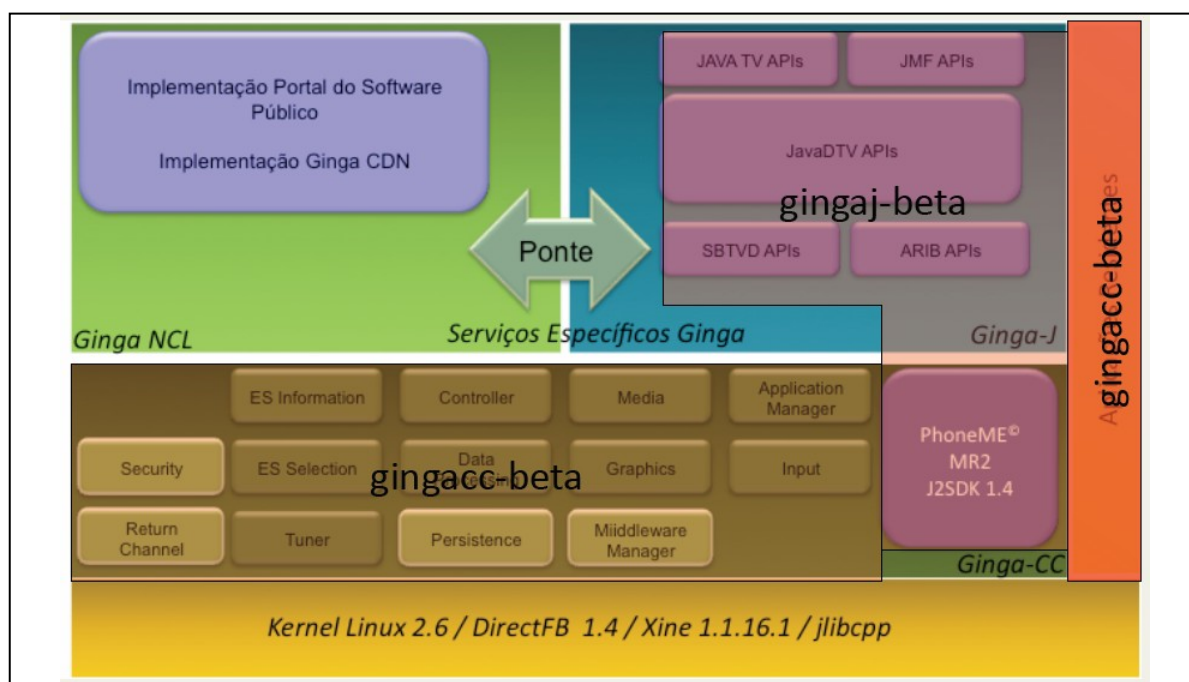


Figura 43. Arquitetura do OpenGinga *release* 1, contendo as bibliotecas Java DTV
 Fonte: KULESZA, R.; FERREIRA, J. (2009, p. 30)

Concluídos os estudos relacionados à arquitetura genérica de um sistema de televisão digital, o mecanismo de interatividade na TV Digital, adentrando no padrão brasileiro e por fim, abordando o *middleware* Ginga e seus subsistemas, dá-se seguimento com apresentação de trabalhos relacionados à temática da pesquisa; logo após, o relato das atividades de projeto, modelagem e desenvolvimento do protótipo proposto como objetivo.

6 TRABALHOS CORRELATOS

Neste Capítulo serão apresentados alguns trabalhos que descrevem o desenvolvimento de aplicações ou protótipos interativos, baseados no paradigma procedural., bem como um trabalho desenvolvido anteriormente nesta instituição baseado no paradigma declarativo do *middleware* Ginga.

6.1 PROTÓTIPO DE JOGO EDUCACIONAL QUIZ EDTEC

Pinto, Queiroz-Neto e Silva (2007) descrevem o desenvolvimento de um protótipo de jogo educacional para TV Digital, denominado “Quiz Edtec”, consistindo em um jogo simples de perguntas e respostas de conteúdo relacionado ao curso de Sistemas de Informação do Centro de Educação Federal do Amazonas, de modo a servir como ferramenta de auxílio ao aprendizado. O jogo foi desenvolvido sobre uma plataforma Java, consistindo em Xlets executados em um emulador de ambiente de execução de TV Digital, sendo que a interação entre o usuário e a aplicação se deu pela utilização do controle remoto do emulador, conforme captura de tela ilustrada na Figura 44:

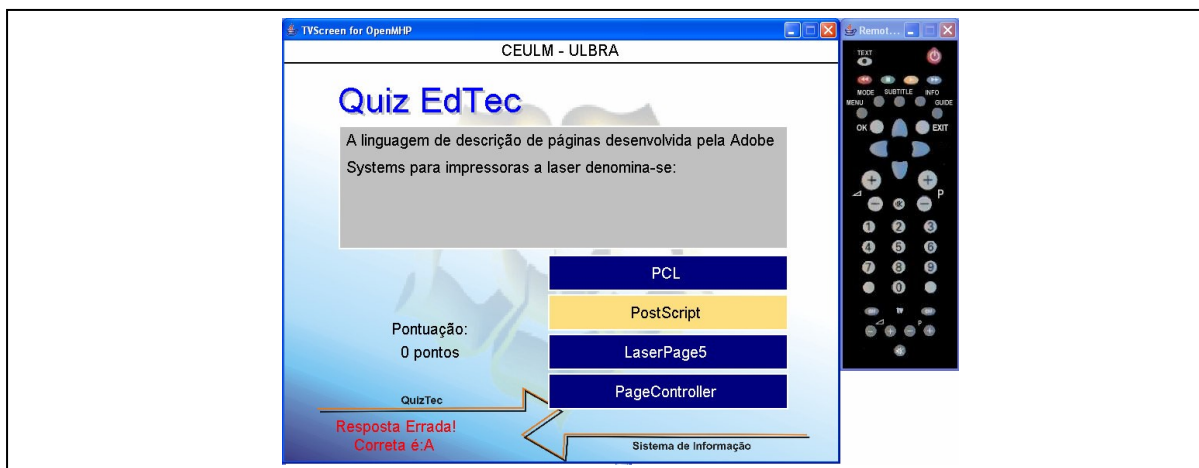


Figura 44. Interface do protótipo de jogo educacional Quiz Edtec
Fonte: PINTO, L.; QUEIROZ-NETO, J.; SILVA, K. (2007, p. 09)

6.2 PROTÓTIPO DIGA DOCTOR

Silva Santos et al (2009) descrevem o desenvolvimento do protótipo DIGA *Doctor*, módulo demonstrativo do projeto DIGA GINGA, idealizado pela Universidade Federal do Ceará como um projeto destinado a utilizar a estrutura do *set-top box* e *middleware* Ginga como ambiente de desenvolvimento de aplicações voltadas ao cidadão em sua residência, como integração entre os dispositivos eletrônicos residenciais, monitoramento de sinais vitais dos cidadãos e outras aplicações neste sentido. O protótipo foi desenvolvido sobre o ambiente Ginga-J e implementado em um ambiente de Computador Pessoal (PC), utilizando-se emuladores e um visualizador de Xlets para simular o funcionamento do sistema. A Figura 45 apresenta a interface do protótipo:

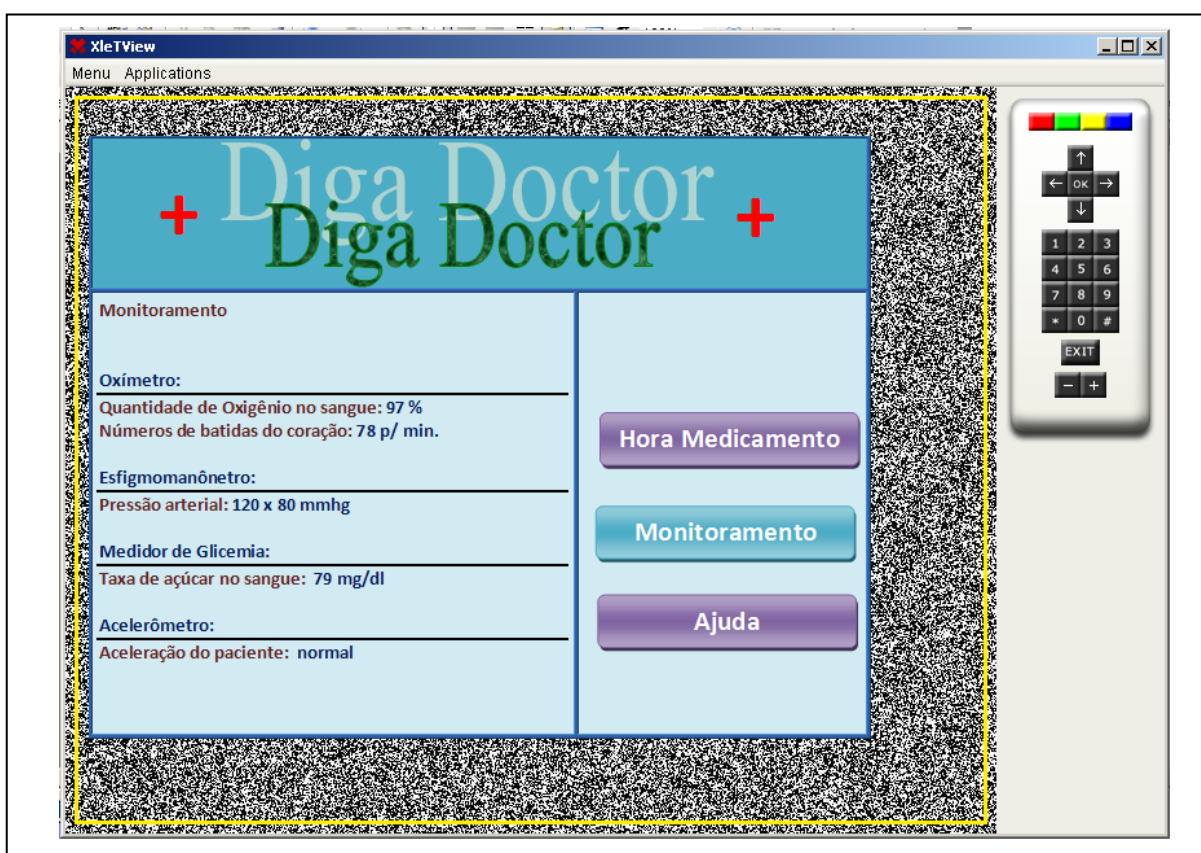


Figura 45. Interface do protótipo Diga Doctor
Fonte: SILVA SANTOS, M. et al. (2009, p. 07)

6.3 PROTÓTIPO T-COMMERCE BASEADO EM MHP E JAVA TV (DARÓS, 2004)

Darós (2004) apresentou como requisito para a obtenção de seu bacharelado um protótipo de aplicação de comércio eletrônico televisivo baseado no *middleware* MHP, do padrão europeu. A exemplo dos outros dois trabalhos, o autor utilizou um ambiente emulador de aparelho receptor para simular o funcionamento da aplicação e sua interação com o usuário (baseado em Java TV), onde este poderia fazer compras fictícias pelo televisor, utilizando o controle remoto como dispositivo de interação.



Figura 46. Interface do protótipo de T-Commerce
Fonte: DARÓS, J. (2004, p. 48)

6.4 PROTÓTIPO CONTROLE DE REDE DOMÉSTICA GINGA HOME

Oliveira (2010) apresenta um estudo de caso envolvendo os subsistemas GINGA-NCL e GINGA-J, onde uma aplicação de controle de dispositivos integrados em uma rede

doméstica foi desenvolvida utilizando os dois paradigmas. Consistia de um portal por onde diversos dispositivos instalados em uma residência fictícia poderiam ser controlados remotamente via aparelho televisor, visto que integrados por uma rede HAN. O aparelho televisor (ou emulador do ambiente receptor) funcionaria conectado a um servidor controlador dos equipamentos da rede (conectados a este via padrão *Sockets*), sendo o controle realizado a partir da interface disponibilizada ao usuário na tela do receptor.

As aplicações procedurais basearam-se no *middleware* MHP e foram executadas com a utilização de emuladores, assim como nos demais trabalhos, enquanto as declarativas basearam-se em NCL e Lua. Ao final, o autor traça um comparativo entre os dois paradigmas, apresentando as vantagens e desvantagens de cada um. A Figura 47 apresenta a interface do programa:



Figura 47. Interface do protótipo de portal de controle de dispositivos residenciais
Fonte: OLIVEIRA, B. (2010, p. 32)

6.5 PROTÓTIPO BASEADO NO SUBSISTEMA GINGA-NCL (DUARTE, 2009)

Duarte (2009) descreve o desenvolvimento de um protótipo simples, com a finalidade de demonstrar a possibilidade de se construir aplicações interativas baseadas no

paradigma declarativo do *middleware* Ginga. O protótipo consiste em permitir ao usuário alternar entre canais de vídeos exibidos no lado receptor, demonstrando a funcionalidade de multiprogramação. O usuário utiliza os botões coloridos do controle remoto para alternar entre um ou outro canal. O protótipo foi desenvolvido em linguagem NCL, sendo utilizado para execução a ferramenta Ginga-NCL Emulator v.1.1.1. A Figura 48 apresenta a interface do protótipo sendo executada no emulador:

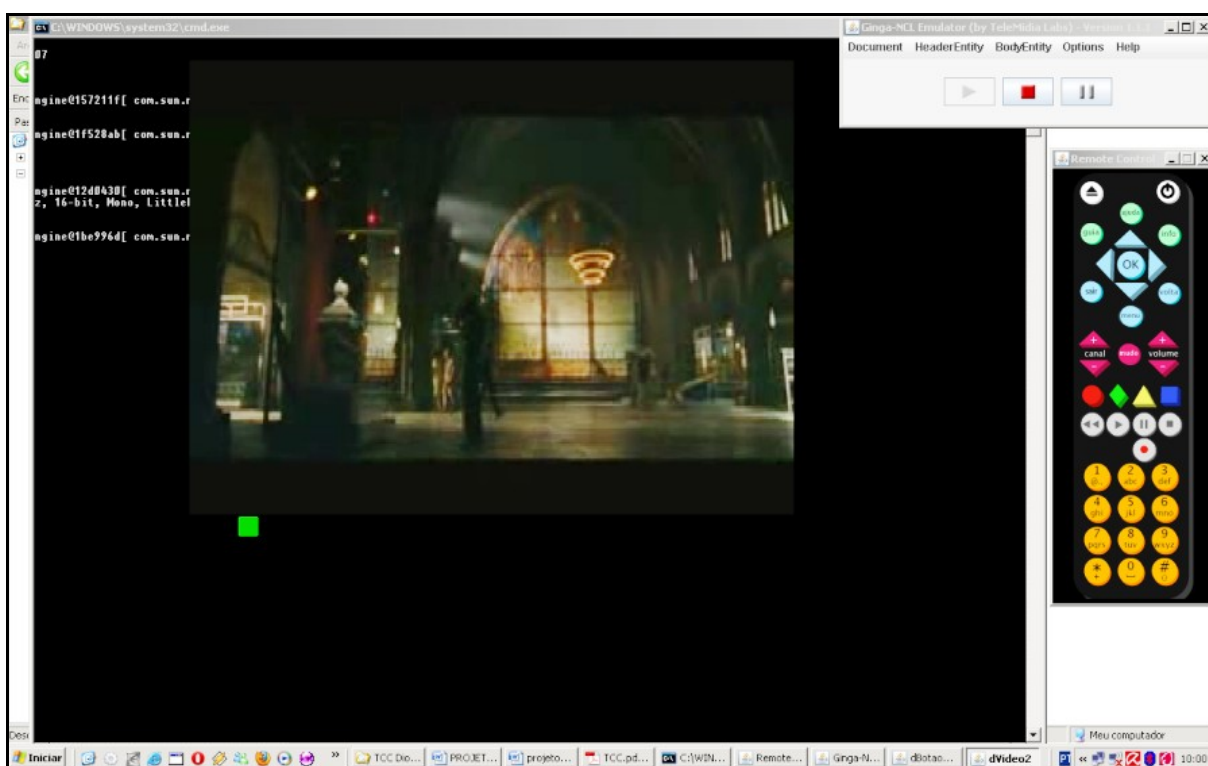


Figura 48. Interface do protótipo de aplicação baseada no subsistema Ginga-NCL
Fonte: Adaptado de DUARTE, D. (2009)

O próximo capítulo apresenta um relato das etapas percorridas na construção do protótipo proposto como objetivo principal desta pesquisa, desde a coleta e análise de requisitos até a fase de testes realizados.

7 PROTÓTIPO DE APLICAÇÃO T-COMMERCE BASEADO EM GINGA-J

O protótipo descrito no objetivo geral do projeto será desenvolvido com base no ambiente procedural do *middleware* Ginga (Ginga-J). A escolha deste subsistema em detrimento ao Ginga NCL levou em conta as características oferecidas pela linguagem e plataforma Java no que diz respeito ao desenvolvimento de aplicações complexas, distribuídas e com interface rica, entre outras características já conhecidas a respeito desta plataforma e suas ferramentas, comentadas anteriormente no Capítulo 1. Levou-se em consideração também o fato de que o subsistema NCL já tenha sido abordado anteriormente nesta instituição, no trabalho desenvolvido por Duarte (2009), o que torna interessante o desenvolvimento de um projeto baseado em tecnologia diversa – neste caso, o subsistema Ginga-J.

Além do mais, se antes a indefinição sobre a adoção ou não das bibliotecas proprietárias do núcleo GEM era um empecilho à adoção do subsistema Ginga-J, a substituição destas pela especificação Java DTV da Sun *Microsystems* reacendeu o interesse pela utilização do subsistema procedural do *middleware* brasileiro, embora a implementação das bibliotecas desta nova especificação venha sendo feita de forma gradativa pela equipe do projeto Ginga *Code Development Network* (Ginga CDN), em um trabalho envolvendo um número considerável de colaboradores e que vêm sendo liberado gradativamente à comunidade.

O protótipo desenvolvido será baseado nas modalidades Internet na TV e *Walled-Garden*, apresentadas no Capítulo 3. Optou-se pelo desenvolvimento de uma aplicação *t-commerce*, onde o usuário poderá efetuar compras pela televisão (neste caso, o ambiente emulador) a exemplo dos sites de comércio eletrônico existentes na Internet, interagindo com o sistema utilizando o controle remoto. Em suma, o usuário poderá navegar entre itens organizados em categorias, adicionar ao carrinho de compras e encerrar a transação.

Por se tratar de um protótipo de aplicação, a intenção é demonstrar a possibilidade de se construir aplicações em Java para TV Digital, mesmo que em ambiente emulador, que possam estabelecer comunicação com serviços remotos e oferecer interatividade ao usuário, utilizando a Internet como meio de comunicação em um ambiente baseado no modelo de comunicação *http Request-Response*. Considerando a troca de informações sigilosas presente neste tipo de aplicação, buscou-se implantar um mecanismo de criptografia simples para demonstrar a possibilidade de se agregar mecanismos de segurança no ambiente.

7.1 METODOLOGIA UTILIZADA E MODELAGEM DO PROTÓTIPO

Durante a fase de levantamento bibliográfico, foi realizado um estudo sobre o ambiente da televisão digital, suas características e funcionamento, bem como a interatividade e sua inserção na televisão digital. Na sequência, foi abordado o Sistema Brasileiro de Televisão Digital, seu histórico e padrão resultante, servindo como prelúdio para o estudo do *middleware* Ginga e seu ambiente procedural, o Ginga-J.

Após o levantamento de algumas aplicações e protótipos correlatos, deu-se continuidade ao projeto, com o desenvolvimento do protótipo utilizando-se a implementação de referência OpenGinga e suas ferramentas.

O desenvolvimento da implementação atualmente está sob os cuidados do projeto Ginga-CDN, vinculado ao laboratório LAViD da Universidade Federal da Paraíba. Além da implementação de referência, concebida para ser executada em ambiente Linux e necessitando compilação após ser baixada do repositório Ginga CDN, o projeto disponibiliza duas ferramentas alternativas para usuários/desenvolvedores (GINGA CDN, 2009):

- a) ginga-J Emulator: funciona em conjunto com a IDE Eclipse proporcionando um ambiente de desenvolvimento e emulação do *middleware* compatível

também com o ambiente Windows;

- b) máquina virtual Ginga: compilação personalizada da distribuição Linux Ubuntu contendo o OpenGinga já instalado e configurado para ser executado pela ferramenta *VirtualBox*, dispensando assim as etapas de download e compilação da implementação de referência, além de permitir a execução a partir de qualquer sistema operacional.

O modelo prescritivo de processo adotado para nortear o desenvolvimento da aplicação protótipo foi o Iterativo Incremental. Este modelo caracteriza-se pela combinação de elementos do modelo em cascata aplicados de maneira iterativa, onde os elementos são desenvolvidos, validados e incluídos no sistema de modo a agregar um novo requisito ao conjunto - isto é, um incremento (PRESSMAN, 2006, p. 41). A Figura 49 ilustra o funcionamento do modelo incremental no decorrer do tempo:

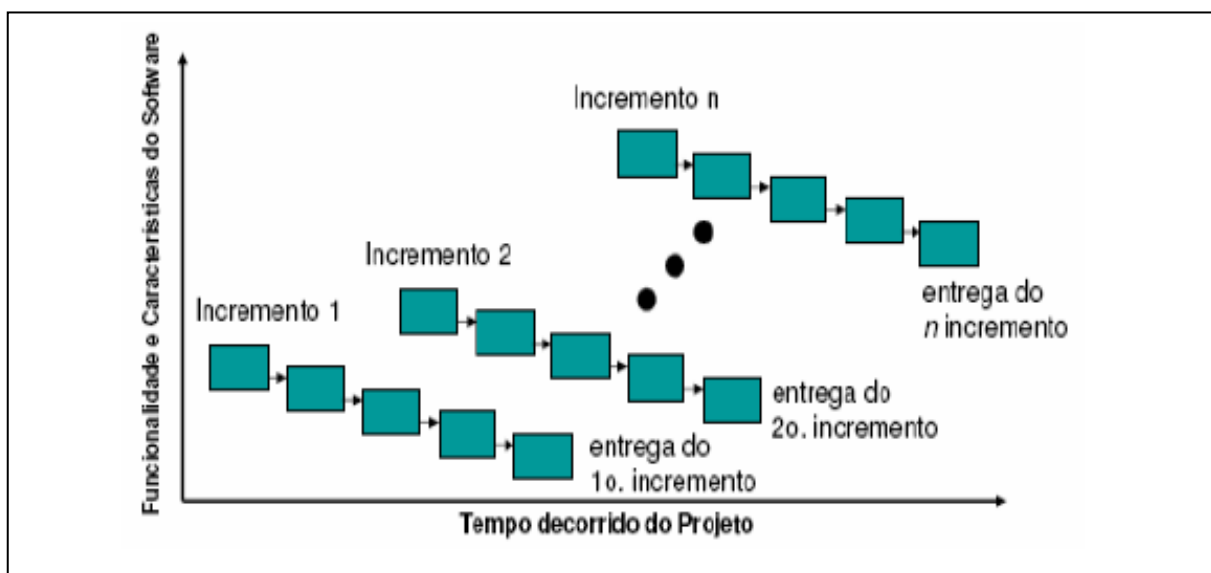


Figura 49. Evolução do desenvolvimento de acordo com o modelo Iterativo Incremental
Fonte: HABBEMA, H. (2009)

Neste modelo, em um primeiro instante são satisfeitos os requisitos básicos do sistema, dando origem ao núcleo do produto, que é validado e revisado. Este então é analisado

e um plano de desenvolvimento é montado com base na análise do que já está pronto, para definir as características que serão implementadas no próximo incremento ou ciclo, gerando um novo produto operacional após o fim do incremento, que será testado e analisado, dando início a um novo ciclo até que todos os elementos e características sejam agregados e o produto completo produzido (PRESSMAN, 2006, p. 41) conforme o modelo ilustrado na Figura 50.

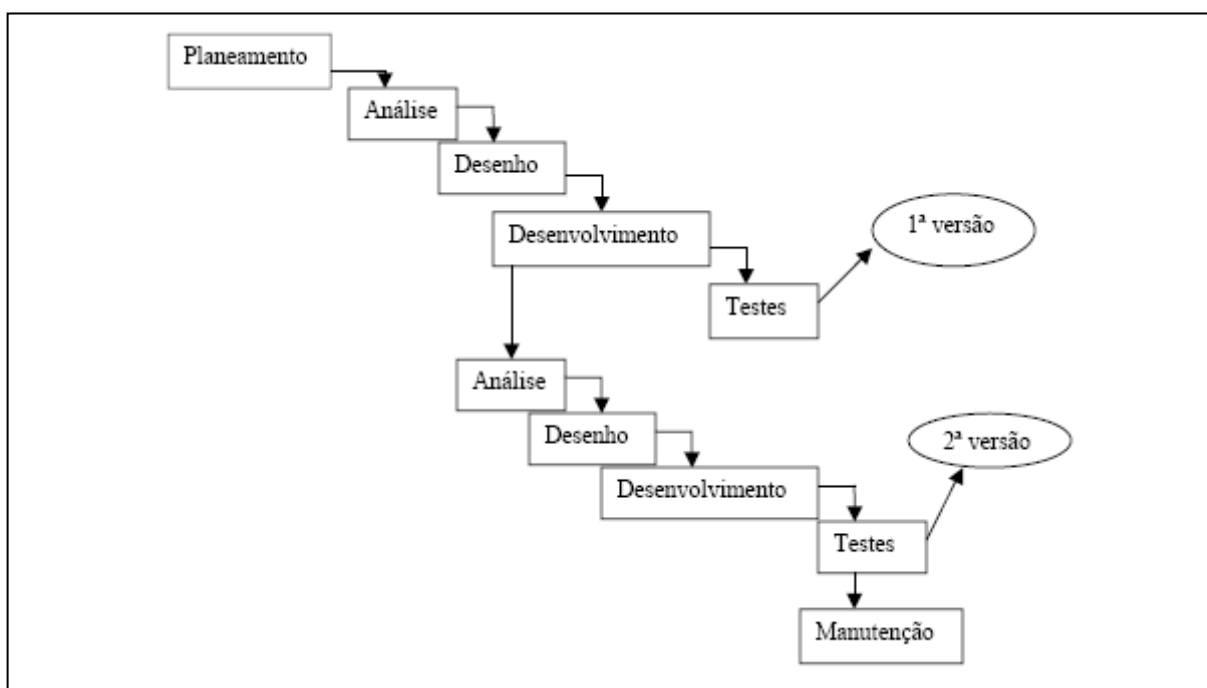


Figura 50. Modelo Iterativo Incremental e seu ciclo de desenvolvimento progressivo
Fonte: Adaptado de CICLO (2006)

Desse modo, o sistema é construído pouco a pouco, sendo satisfeitos a cada iteração os requisitos mais críticos que ainda não foram agregados, em uma evolução gradativa, gerando versões mais refinadas e detalhadas do produto a cada iteração. Este modelo é útil quando não se dispõe de mão de obra suficiente para entregar o produto completo dentro do prazo estipulado, permitindo a entrega de versões iniciais mais robustas ao usuário, e posterior agregação de novas funcionalidades.

7.1.1. Funcionamento do sistema e identificação de requisitos

Embora o desenvolvimento do protótipo não esteja inserido em um cenário real, onde técnicas da Engenharia de Requisitos seriam utilizadas na obtenção de detalhes do funcionamento do sistema, identificação e análise dos requisitos, definiu-se um cenário hipotético para obtenção de informações acerca do sistema a ser projetado e desenvolvido.

Neste cenário hipotético, um *Workshop* de Requisitos envolvendo representantes do negócio e o analista serviria como ferramenta para levantamento do funcionamento desejado do protótipo e um esboço de sua interface gráfica, identificando-se a partir destes os requisitos básicos do sistema, como segue:

- a) o usuário utilizará o controle remoto para navegar entre as categorias e itens disponíveis para compra;
- b) uma vez escolhida uma categoria no painel de categorias, os itens correspondentes são exibidos em um painel de navegação horizontal, para que o usuário percorra e seleciona um item para ser exibido de maneira detalhada;
- c) ao selecionar um item, detalhes deste são exibidos em um painel específico;
- d) o usuário poderá então adicionar itens em um carrinho de compras;
- e) se o usuário não estiver autenticado no servidor remoto, no momento em que adicionar um item no carrinho pela primeira vez, será solicitado o seu *login*;
- f) caso não esteja cadastrado na loja virtual como cliente, será informado um site o qual deverá acessar e informar seus dados cadastrais, visto que exigir um cadastro completo via controle remoto, sem o uso de um teclado com letras, dificulta a interação;
- g) a página de cadastro funcionará em ambiente web independente (à parte) do ambiente/emulador de televisão (para facilitar o processo de cadastro);

- h) em tal página, o cliente informará os dados exigidos, inclusive senha (numérica, para facilitar a interação) e receberá um código de usuário numérico, o qual utilizará para se autenticar no momento da compra;
- i) uma vez logado, alguns dados do usuário serão exibidos na tela da aplicação para conferência;
- j) no decorrer do processo, uma listagem com os itens previamente marcados para compra deverá estar acessível ao usuário, para possível exclusão de itens, confirmação da operação ou retornar e selecionar outros itens para compra.

A Figura 51 traz um esboço da interface principal da aplicação, que poderia ter sido definida pelos representantes do cliente durante a coleta de informações:

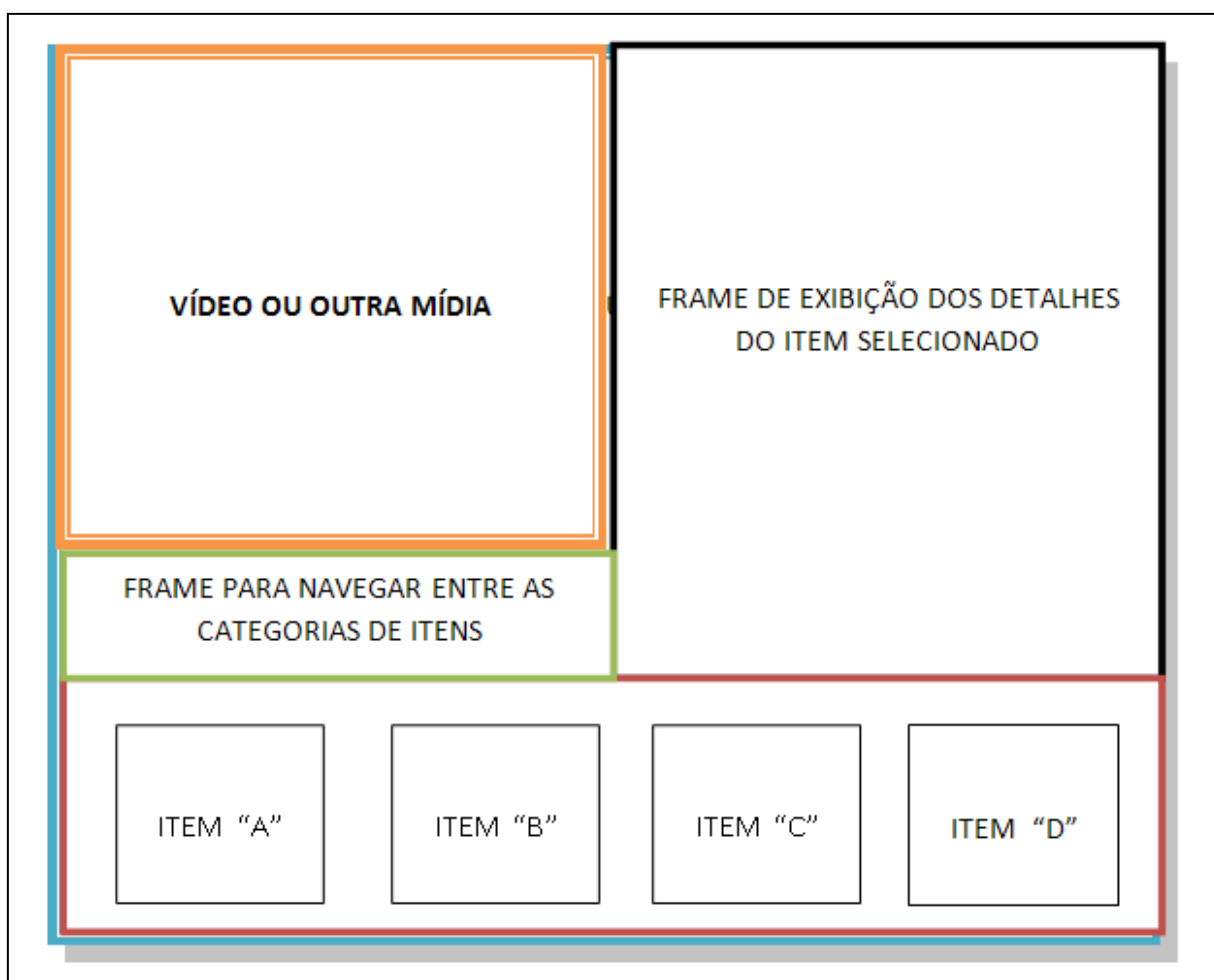


Figura 51. Esboço da interface da aplicação definida no Workshop hipotético

7.1.2. Análise e levantamento de requisitos

Uma vez realizada a etapa de obtenção de dados junto ao cliente/usuário, passa-se à etapa de análise destas informações e levantamento de requisitos do protótipo a ser desenvolvido. A Tabela 9 apresenta os requisitos funcionais levantados:

Tabela 9. Lista de Requisitos Funcionais levantados (continua)

REQUISITO	DESCRIÇÃO	OCULTO	EVIDENTE
F01: Exibir listagem de categorias	O sistema deverá exibir em um painel específico as categorias de produtos, para que o usuário navegue com o controle remoto e selecione a categoria desejada, para listagem dos respectivos itens.		X
F02: Exibir Listagem de itens	O sistema deverá retornar a listagem de itens da categoria selecionada pelo usuário. Os itens em descrição e ícone resumido serão dispostos em um painel específico, que proporcione a navegação do usuário pela listagem e posterior seleção de item.		X
F03: Exibir detalhes do item	Ao selecionar um item no painel de itens exibidos, um painel específico exibirá o produto com descrição, características e imagem ampliada.		X
F04: Alternar entre categorias e itens	O usuário poderá alternar entre categorias de itens e entre itens de uma categoria listada, utilizando os botões do controle remoto.		X
F05: Adicionar itens no carrinho	O usuário poderá adicionar o item visualizado ao carrinho de compras.		X
F06: Autenticação de usuário	O usuário deverá autenticar-se para poder adicionar itens ao carrinho de compras. Não estando autenticado no momento da adição de um item ao carrinho, será encaminhado à tela de Login/autenticação.		X
F07: Visualizar carrinho de compras	O usuário poderá consultar no carrinho de compras os itens marcados para compra e fazer exclusões, caso necessário.		X
F08: Efetuar compra	O usuário poderá efetuar a compra dos itens constantes no carrinho. Nesse instante, uma tela exibirá os itens selecionados, os dados do usuário e da compra.		X

Tabela 9. Lista de Requisitos Funcionais levantados (continuação)

REQUISITO	DESCRIÇÃO	OCULTO	EVIDENTE
F09: Confirmar transação	O usuário poderá confirmar a compra, acionando o botão Confirmar na tela que exibe os itens selecionados e dados da compra e do usuário.		X
F10: Cancelar/zerar carrinho de compra	O usuário poderá cancelar todo o carrinho de compra, bastando acionar a opção correspondente. Nesse momento, o usuário permanecerá autenticado e poderá selecionar outros itens.		X
F11: Desconectar / logout	O usuário será desconectado assim que encerrar a transação ou fechar a aplicação.	X	
F12: Cadastro de usuários	Caso o usuário seja inválido ou não esteja cadastrado, o sistema irá solicitar que efetue seu cadastro, informando o endereço do site/página específica para este fim. Esta operação será feita via navegador, fora do ambiente de aplicação. Ao informar seus dados e concluir o cadastro, um código numérico de usuário será fornecido, para ser utilizado no Login da aplicação de compras.		X

Os requisitos não funcionais levantados estão arrolados na Tabela 10:

Tabela 10. Lista de Requisitos Não Funcionais levantados

REQUISITO	RESTRIÇÃO	CATEGORIA	DESEJÁVEL	PERMANENTE
NF01: Middleware	O ambiente emulado de <i>set-top box</i> exige a instalação da implementação de referência do <i>middleware</i> Ginga – subsistema Ginga-J	Software		X
NF02: Servidor remoto	A aplicação de compras funcionará em conexão com um servidor remoto, em modelo cliente-servidor.	Software / arquitetura		X
NF03: Navegação facilitada	O acionamento dos comandos deverá ser facilitado, devido à utilização dos botões coloridos (funções) e botões numéricos do controle remoto virtual.	Interface	X	
NF04: Status da navegação	O nome do usuário autenticado deverá ser exibido em algum ponto da interface gráfica, bem como a quantidade de itens adicionados ao carrinho e total em \$\$.	Interface	X	

7.1.3. Casos de uso levantados

A Tabela 11 apresenta a listagem de casos de uso encontrados, de forma resumida. O conjunto dos casos de uso em sua forma expandida pode ser visto de maneira detalhada no Apêndice 1 deste trabalho.

Tabela 11. Listagem de casos de uso em forma resumida

NOME	DESCRIÇÃO
C01	Navegar pela aplicação.
C02	Listar itens de uma categoria.
C03	Exibir detalhes de um item específico.
C04	Adicionar itens ao carrinho de compras.
C05	Efetuar login/autenticação.
C06	Consultar carrinho de compras.
C07	Excluir item do carrinho de compras.
C08	Zerar carrinho de compras.
C09	Efetuar a compra dos itens existentes no carrinho.
C10	Efetuar logout.
C11	Cadastro de usuário no sistema de compras (página à parte).

7.1.4. Diagramas de casos de uso

Os casos de uso elementares à operação da aplicação levantados durante o processo de análise e modelagem podem ser representados pelos diagramas expostos a seguir. Enquanto a Figura 52 apresenta os casos de uso relativos ao uso da aplicação Xlet pelo usuário, a Figura 53 ilustra o processo de cadastro do cliente no servidor remoto.

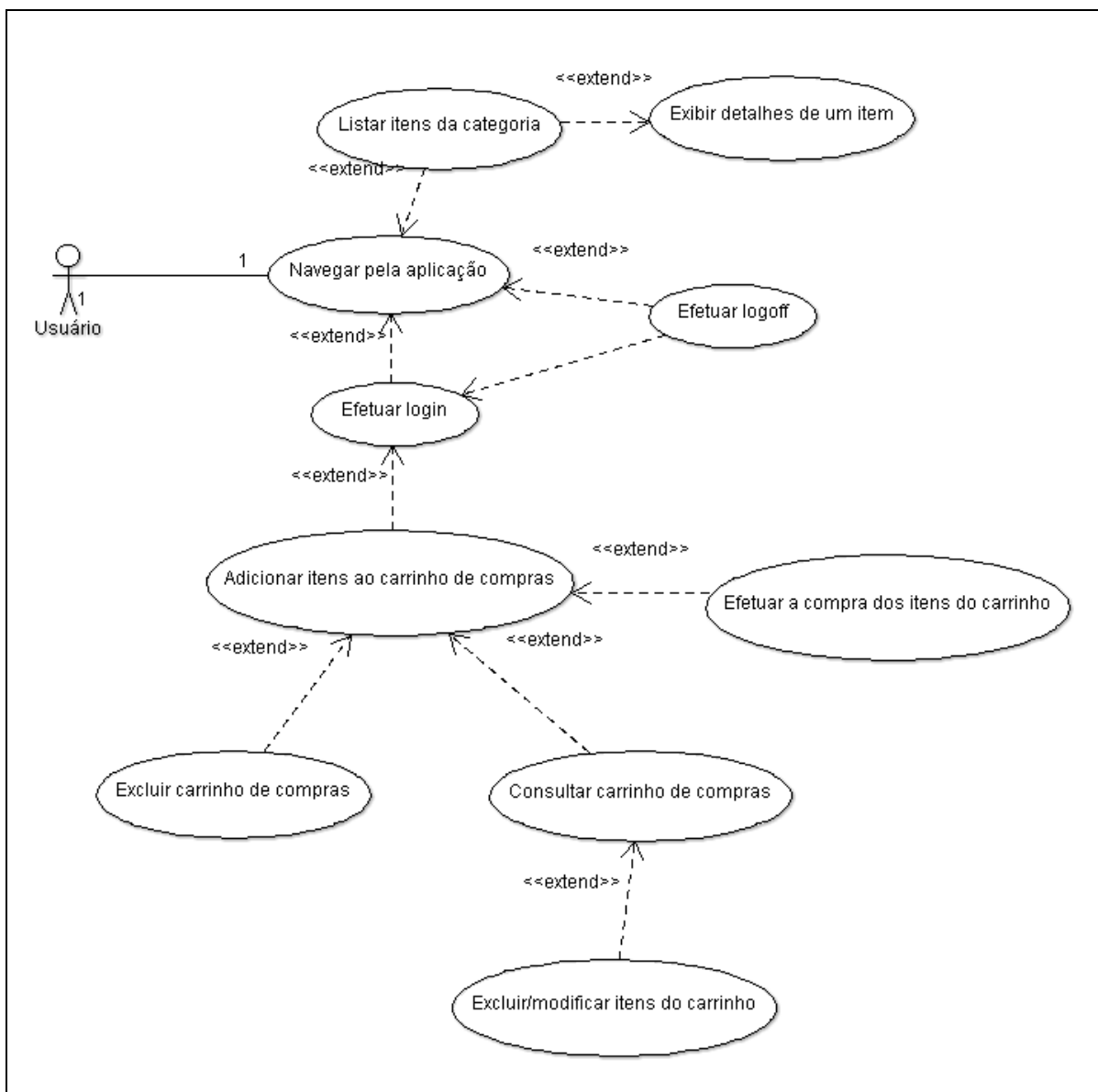


Figura 52. Diagrama de casos de uso da interface do aplicativo T-Commerce

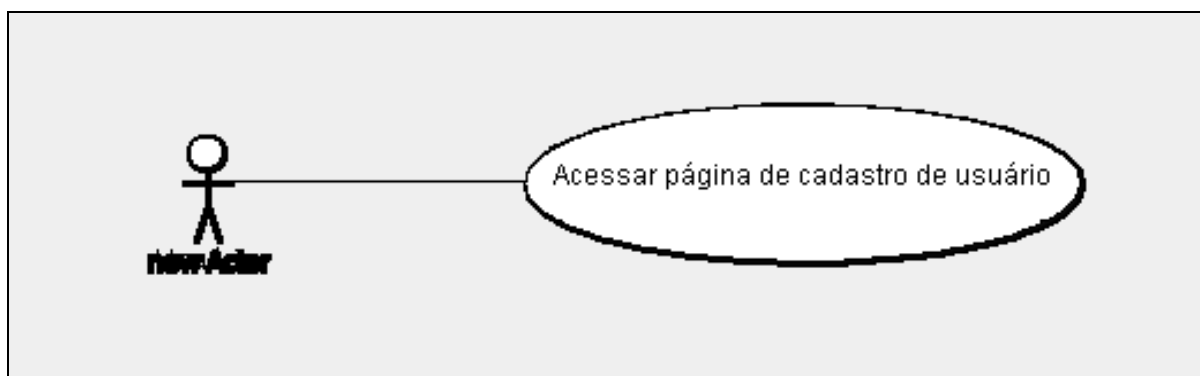


Figura 53. Diagrama de casos de uso do cadastro de usuário/cliente

7.1.5. Matriz de requisitos funcionais x casos de uso

O cruzamento entre os requisitos funcionais levantados e os casos de uso pode ser visto na Figura 54:

RF \ C	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11
F01	X										
F02	X	X									
F03	X		X								
F04	X	X									
F05	X			X							
F06					X						
F07	X					X	X				
F08									X		
F09									X		
F10	X							X			
F11										X	
F12											X

Figura 54. Matriz de requisitos funcionais x casos de uso

7.2 DESENVOLVIMENTO DO PROTÓTIPO

De posse das informações levantadas durante a fase de análise, bem como do mapeamento e diagramas elaborados, passou-se a fase de desenvolvimento da aplicação. Do ponto de vista arquitetônico, o sistema protótipo de *T-Commerce* construído é composto por três partes:

- a) a aplicação Xlet, atuando como interface com o usuário no receptor;

- b) o servidor web remoto, encarregado de receber e tratar as requisições enviadas pela aplicação Xlet;
- c) o canal de retorno/interatividade: uso de um canal de rede/Internet como meio de comunicação entre o cliente e o servidor web.

7.2.1. Servidor web

O servidor de aplicações *web* foi construído com auxílio da ferramenta *NetBeans IDE 6.9 Beta*, em sua versão Java EE. Esta versão da IDE foi escolhida por trazer embutidos os servidores de aplicação *Apache Tomcat 6.0.26* e *GlassFish Server Open Source Edition 3.0.1*.

Basicamente consiste em um sistema *model-view-controller* (MVC) onde a classe *ServerController* atua como *servlet* controlador, recebendo as requisições e direcionando para as ações correspondentes, conforme exposto na Figura 55:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try
    {
        // Trata as requisições enviadas pelo cliente e redireciona..
        if (request.getRequestURI().endsWith("/retornaItem"))
        {
            // Chama o método que retorna um item completo..
            retornaItem(request, response);
        }
        else
        if(request.getRequestURI().endsWith("/retornaCategoria"))
        {
            // Chama o método que retorna os ícones da categoria...
            retornaCategoria(request, response);
        }
        else
        if(request.getRequestURI().endsWith("/autenticar"))
        {
            // Chama o método que autentica o cliente..
            clienteAutenticar(request, response);
        }
        else
        if(request.getRequestURI().endsWith("/efetivar"))
        {
            // Chama o método que autentica o cliente..
            transacaoEfetivar(request, response);
        }
        } ... (continua)...
```

Figura 55. Método *processRequest*, direcionando as requisições no servidor

Como se pode observar na Figura 55, os principais métodos da classe, que processam as requisições e devolvem resposta ao usuário são:

- a) *retornaItem(request, response)*: retorna um conjunto de informações representando um artigo de venda, tratado no sistema como objeto *ItemCompleto*;
- b) *retornaCategoria(request, response)*: retorna o conjunto de Itens de dada categoria de artigo (calçados, por exemplo), com informações resumidas - objeto *Icone*;
- c) *clienteAutenticar(request, response)*: responsável pela autenticação do usuário cliente, retornando seus dados cadastrais caso esteja cadastrado;
- d) *transacaoEfetivar(request, response)*: trata a efetivação da operação de compra, alimentando o banco de dados.

Completando o padrão MVC, há os objetos Java *Beans* na camada de modelo (pacote *modelo*), representando as entidades envolvidas no negócio (*Cliente*, *Icone*, *ItemCarrinho*, *ItemCompleto* e *Transacao*). O pacote *util* reúne, entre outras funcionalidades, os objetos *Data Access Object* (DAO) responsáveis pelas transações com o banco de dados: *ItemDao* e *ClienteDao*.

Para armazenar os dados envolvidos, utilizou-se o sistema gerenciador de banco de dados (SGBD) *Firebird* versão 2.0.4 (*Win32 Build*). Foi criada uma base de dados chamada TCOMMERCE.FDB, no dialeto SQL 3 e conjunto de caracteres WIN1252, que permite o uso de caracteres acentuados e especiais, como ‘ç’, comuns em línguas latinas, de modo especial na língua portuguesa.

O diagrama entidade-relacionamento (E-R) do modelo relacional, contendo as tabelas gerenciadas pelo SGBD do servidor pode ser visto no Apêndice 2.

Para efeitos de demonstração de funcionamento, optou-se pela execução do servidor diretamente na IDE *NetBeans*, nada impedindo entretanto a implantação de uma compilação independente em um servidor autônomo, desde que capaz de atuar como *container* de *servlets* e receber requisições *web* - como o *Apache Tomcat* por exemplo, de forma a operar sem vínculos com o *NetBeans*.

7.2.2. Aplicação Xlet TCommerce

Na fase de desenvolvimento do Xlet protótipo de aplicação *T-Commerce*, elemento principal e meta deste trabalho, utilizou-se o projeto Gingga-J Emulador, emulador desenvolvido pela equipe do projeto Gingga CDN com base no projeto XletView (implementação Java das bibliotecas MHP). O emulador ainda está em fase de desenvolvimento e implementação das bibliotecas *JavaDTV* e *SBTVD*.

Além das funcionalidades de registro e execução de aplicações no receptor virtual, o emulador traz consigo as bibliotecas *Java TV*, *JavaDTV* e os pacotes de inovação *SBTVD*, possibilitando o também o desenvolvimento de Xlets pelo usuário/desenvolvedor, bastando para tanto importar a *library* `gingaj-emulator.jar` que vem inclusa nos pacotes do emulador.

Embora tenha sido utilizado o emulador para o desenvolvimento do protótipo, com algumas modificações pode-se executar a aplicação na implementação de referência OpenGingga, em fase de desenvolvimento pelo projeto Gingga CDN e concebida para executar em ambiente Linux. Na realidade, o objetivo do projeto é o desenvolvimento desta implementação de referência; o emulador é disponibilizado paralelamente apenas como ferramenta para que desenvolvedores iniciantes se habituem à nova tecnologia mais facilmente, visto que basta executá-lo a partir do Eclipse, dispensando as etapas exigidas para

instalação e compilação da implementação de referência propriamente dita – o OpenGinga no ambiente Linux.

O emulador precisa ser importado do repositório do site Ginga CDN ou de um repositório GIT local para dentro da IDE Eclipse. Após algumas configurações, deve ser executado dentro da IDE. Uma vez que a tela do emulador é exibida, as aplicações desenvolvidas devem ser registradas para que possam ser executadas e testadas. No Apêndice 3 é apresentado um passo-a-passo da configuração do emulador e registro de aplicações desenvolvidas, bem como do servidor *web* na IDE NetBeans.

O desenvolvimento foi feito sobre a plataforma CDC-1.1/*Foundation Profile 1.1/Personal Basis Profile 1.1*, plataforma Java adequada ao desenvolvimento de aplicações para dispositivos embarcados e móveis, inclusive Java para TV, conforme ilustrado pela Figura 56:

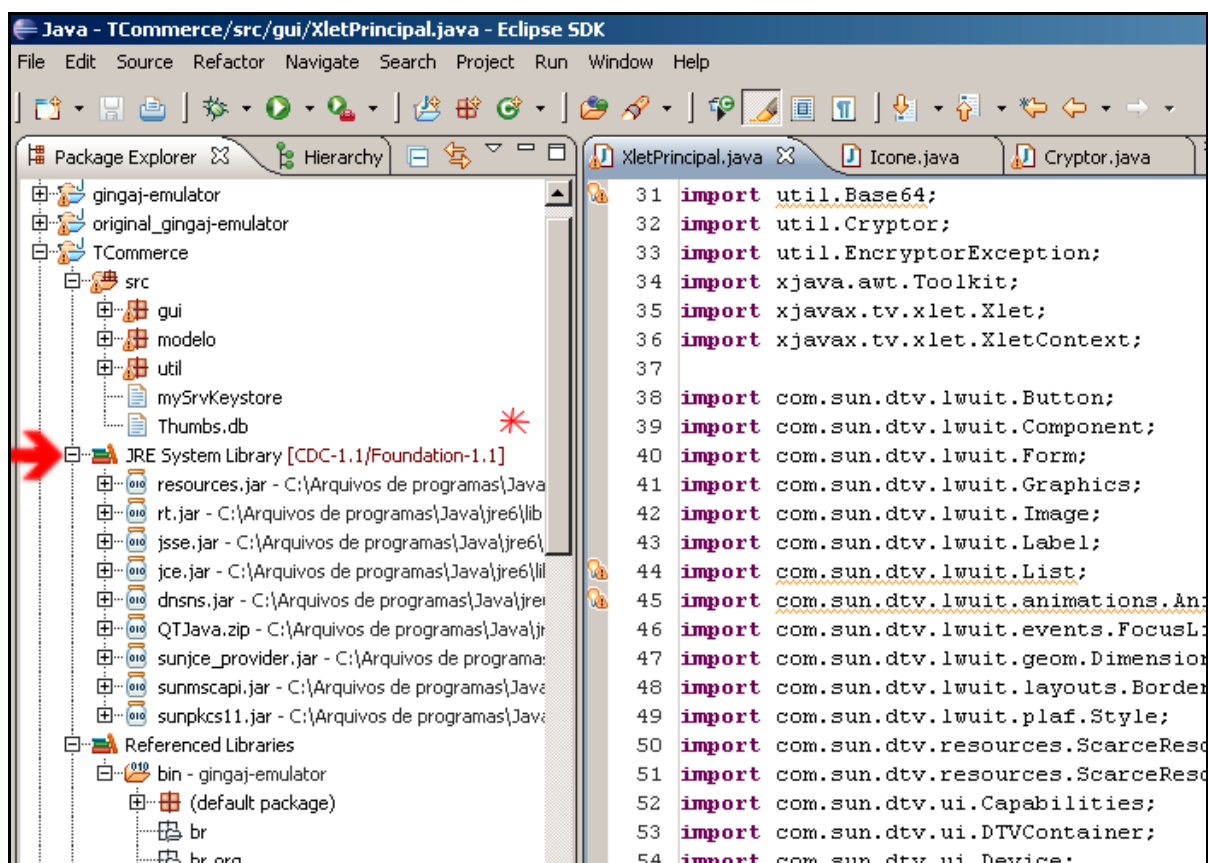


Figura 56. Emprego da plataforma CDC/FP/PBP 1.1 na IDE Eclipse

As bibliotecas base para o desenvolvimento do protótipo, assim como em todas as aplicações desenvolvidas no subsistema Ginga-J, são *JavaTV* e *JavaDTV*. A primeira disponibiliza duas interfaces essenciais à implementação:

- a) *Xlet*: é a interface padrão e base da aplicação, pois a classe principal da aplicação deve implementá-la, sobrescrevendo seus métodos como fora descrito no Capítulo 5;
- b) *XletContext*: responsável por disponibilizar à aplicação informações acerca do ambiente – contexto - no qual está sendo executada, além de servir como ponte de comunicação de mudanças de estado entre a aplicação e o contexto.

JavaDTV fornece uma série de pacotes relacionados com a interface gráfica. Para a construção do protótipo utilizou-se:

- a) *lwuit*: fornece os componentes (*widgets*) avançados de interface com o usuário, como contêineres, formulários, botões, campos de texto, estilos de layout, entre outros;
- b) *ui*: contém classes responsáveis pela manipulação dos elementos do visor, suporte aos eventos de entrada disparados no controle remoto e *widgets* da interface gráfica (ouvintes e tratamento de eventos), manipulação do plano gráfico do dispositivo, gerenciamento dos recursos entre as aplicações concorrente, entre outros.

Uma vez capturado o contexto do ambiente e configurado o dispositivo de exibição (Figura 57), para a exibição da aplicação foram seguidos os seguintes passos:

- a) criação os elementos *LWUIT* (*Form* e componentes gráficos contidos nele);
- b) obtenção de um contêiner - *DTVContainer*;
- c) inserção do elemento gráfico principal (*Form*) no contêiner obtido;
- d) exibição da interface criada.

```

// Captura o contexto da aplicação..
this.context = context;

/* Uso do JavaDTV UI - Configurações da tela */

Device.getInstance();
Screen currentScreen = Screen.getCurrentScreen();

// Configuração dos planos gráficos
Plane[] planes = currentScreen.getAllPlanes();

for( int i=0; i < planes.length; i++ )
{
    Capabilities cap = planes[i].getCapabilities();
    if (cap.isGraphicsRenderingSupported())
    {
        plane = planes[i];
        planeSetup = plane.getCurrentSetup();
        break;
    }
}

try
{
    plane.reserve(false, -1, this);
} catch (Exception e)
{
    e.printStackTrace();
}

// Cria um template para configurar o plano gráfico
PlaneSetupPattern template = new PlaneSetupPattern();

// Configuração com suporte a escalonamento de imagens..
template.setPreference(
    PlaneSetupPattern.IMAGE_SCALING_SUPPORT,
    PlaneSetupPattern.PREFERRED);

// Garante que aplicações em execução não sejam afetadas pelas configurações..
template.setPreference(
    PlaneSetupPattern.NO_GRAPHICS_IMPACT,
    PlaneSetupPattern.REQUIRED);

// Obtem uma configuração de plano compatível com os parâmetros..
PlaneSetup setup = plane.getBestSetup(template);

// Agora aplica as configurações escolhidas..
if(setup != null)
{
    try
    {
        // Reserva o plano gráfico..
        plane.reserve(false, -1, this);
        plane.setPlaneSetup(setup);

        // Libera o plano gráfico..
        plane.release();
    }
    catch(Exception e)
    {
        //
        e.printStackTrace();
    }
}
}

```

Figura 57. Trecho do método *initXlet* -captura do contexto e configuração do dispositivo

Na Figura 58 observa-se o processo de criação do componente gráfico de mais alto nível – o *Form* – bem como a criação dos demais *widgets* e adição ao formulário.

```

/* Uso do JavaDTV UI - Tratamento de eventos */
UserInputEventManager manager
    = UserInputEventManager.getUserInputEventManager(currentScreen);
RemoteControlEvent anyColoredKeyTyped
    = new RemoteControlEvent( null, java.awt.event.KeyEvent.KEY_TYPED,
        0, 0,
        RemoteControlEvent.VK_COLORED,
        KeyEvent.CHAR_UNDEFINED);

try
{
    anyColoredKeyTyped.reserve(true, -1, null);
}
catch (Exception e)
{
    e.printStackTrace();
}

// Registra um ouvinte para os eventos do controle remoto..
manager.addUserInputEventListener(this, anyColoredKeyTyped);

/* Uso do JavaDTV LWUIT - Criando componentes */
// Cria o formulário principal
form = new Form();
form.setLayout(null);
form.getContentPane().setLayout(null);
form.setSize(planeSetup.getScreenResolution());
form.getStyle().setBgColor(Color.WHITE);
form.setTintColor(COLOR_BLUE);

// Label Carrinho de compras..
Image image_carro;
try
{
    image_carro =
        Image.createImage("carro.png");
    lb_carro_icone = new Label();
    lb_carro_icone.setIcon(image_carro);
    lb_carro_icone.setX(lb_fundo_topo_item.getX() + 5);
    lb_carro_icone.setY(lb_fundo_topo_item.getY() + 3);
    lb_carro_icone.setSize(new Dimension( image_carro.getWidth(),
        image_carro.getHeight()));
}
catch (IOException e1)
{
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
// Adiciona o componente Carrinho de Compras ao Form..
form.addComponent(lb_carro_icone);

```

Figura 58. Criação do componente gráfico principal e *widgets* contidos por ele

Neste ponto se faz uma ressalva: na elaboração do protótipo, além da classe *Form*, utilizou-se somente o componente gráfico *Label* na montagem da interface gráfica, porque os

demais *widgets* – como *Button*, *TextField*, *TextArea* e outros – ainda estão em fase de implementação pela equipe do Ginga CDN, estando indisponíveis para utilização até então. No Anexo B pode-se conferir um relatório de *Status* de Implementação dos pacotes *LWUIT* até o momento da elaboração do protótipo.

No método sobrescrito *startXlet()*, obtém-se o contêiner do plano selecionado, representado por um objeto da classe *DTVContainer*. Em seguida, o componente *Form* criado no método *initXlet()* - contendo os demais componentes da interface com o usuário – é adicionado ao contêiner e a interface da aplicação é então exibida no visor. A Figura 59 ilustra este processo:

```

public void startXlet()
{
    System.out.println("Apresentando GUI...");

    // Primeiro obtemos o Container do plano gráfico selecionado..
    dtvcontainer = DTVContainer.getDTVContainer(plane);
    dtvcontainer.setLayout(new BorderLayout());
    dtvcontainer.setSize(planeSetup.getScreenResolution());

    // Adicionando o Form no DTVContainer
    dtvcontainer.addComponent(BorderLayout.CENTER, form);
    dtvcontainer.setVisible(true);

    // Apresentando os componentes da tela
    form.repaint();
    dtvcontainer.repaint();
    dtvcontainer.revalidate();

    // Carrega as instruções de uso..
    instrucoesUsoCarregar();

    // Chama o método que carrega as ofertas..
    ofertasCarregar();
}

```

Figura 59. *startXlet*: captura do contêiner, adição do formulário e exibição da interface

Considerando-se que o protótipo, caso fosse uma aplicação real, seria executado em um ambiente televisivo, projetou-se a interface de modo que toda a interação com o usuário é realizada com o uso do controle remoto, utilizando-se as teclas numéricas disponíveis no controle remoto – de modo especial as teclas coloridas de interatividade. A

Figura 60 ilustra a execução do protótipo na tela do emulador, exibindo a tela inicial contendo promoções e um guia de Instruções de uso das teclas do controle remoto, para facilitar o processo de navegação na interface:

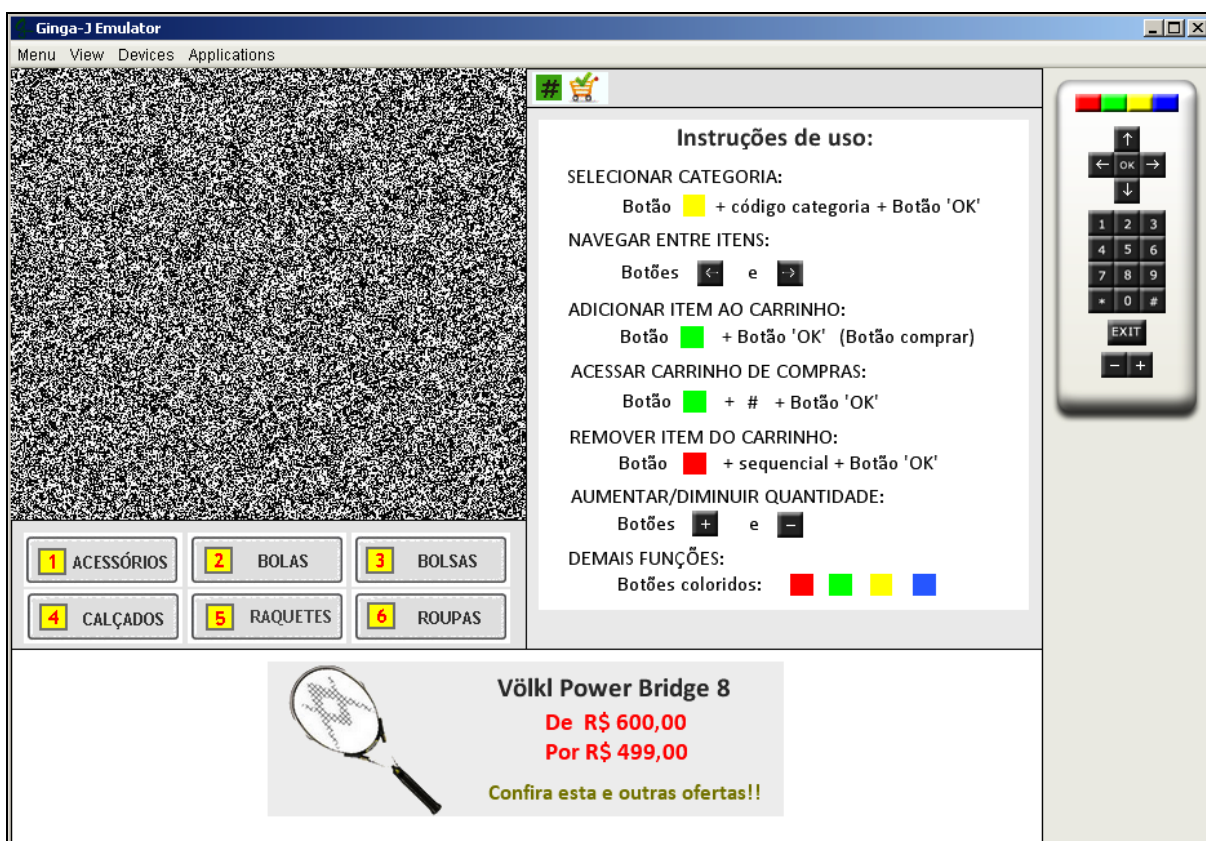


Figura 60. Tela inicial do protótipo T-Commerce, com as instruções básicas de uso

Em virtude de ainda não estarem disponíveis para utilização componentes *widjets* como botões, áreas de texto, caixas de listagem, todos os elementos gráficos dispostos no frame foram construídos com componentes do tipo etiqueta – classe *Label* – o que dificulta o processo de interação entre o usuário e a interface, visto que tal componente não dá suporte a captura de ações/eventos. A solução encontrada foi a criação de um mecanismo de disparo de ações envolvendo as teclas coloridas do controle remoto - funcionando como gatilhos - em associação com um conjunto de variáveis booleanas atuando como chaves de estados exclusivos da aplicação – bloqueio em semáforo.

Um exemplo, dentre outros ilustrados na Instrução de Uso da Figura 60, é a seleção de uma categoria de produtos para exibição no painel inferior: pressionando-se a tecla amarela do controle é ativado o estado *seleção de categoria*, onde a variável *ib_selecao_categoria* assume o valor *true*, e as demais chaves assumem o valor *false*. A partir deste ponto, o ouvinte de eventos armazena todas as teclas pressionadas até que a tecla <OK> seja pressionada, encerrando o processo de captura do código de categoria desejado e chamando o método que requisita as informações da categoria junto ao servidor web remoto; e assim sucessivamente, para cada funcionalidade correspondente aos casos de uso identificados durante o projeto da interface. A Figura 61 ilustra a inicialização das variáveis de controle, no método *initXlet()*:

```
public void initXlet( XletContext context )
{
    System.out.println( "Inicializando o Xlet..." );

    // Inicializa as variáveis de controle..
    ib_selecao_categoria      =      false;
    ib_selecao_item          =      false;
    ib_item_selecionado      =      false;
    ib_botao_verde           =      false;
    ib_botao_vermelho        =      false;
    ib_botao_amarelo         =      false;
    ib_botao_avancar         =      false;
    ib_botao_retroceder      =      false;
    ib_mostrando_carrinho    =      false;
    ib_mostrando_login       =      false;
    ib_mostrando_login_invalido
                             =      false;
    ib_mostrando_compra_final
                             =      false;
    ib_usuario_logado        =      false;
    ib_ofertas_exibir        =      true;

    ... (continua) ...
}
```

Figura 61. Inicialização das variáveis de controle (chaves de estado) da aplicação

Dos casos de uso levantados na Tabela 11, os casos C02 (listar itens de uma categoria), C03 (exibir detalhes de um item), C05 (efetuar *login*) e C09 (efetuar a compra) dependem totalmente do funcionamento do mecanismo de interatividade remota – o uso do canal de retorno (Internet) para requisição de dados a um servidor *web* remoto. A montagem

dos itens exibidos nos painéis em cada um destes casos depende dos dados recebidos em resposta à requisição HTTP feita ao *web service* T-Commerce Server, apresentado na seção 7.2.1.

Neste momento entram em cena os pacotes *java.net* e *java.io*. Enquanto o primeiro contém classes que implementam os mecanismos de comunicação via rede entre aplicações e serviços remotos – tais como requisições HTTP URL, *sockets* e datagramas, entre outros – o segundo contribui com as classes *ObjectOutputStream* e *ObjectInputStream*, responsáveis pela escrita e leitura de objetos serializados em uma *stream* de saída/entrada. O uso dessas classes é fundamental ao funcionamento da aplicação, pois suporta a *passagem de objetos* via requisição HTTP entre o servidor e a aplicação cliente ou vice-versa, facilitando o trabalho de leitura-escrita de parâmetros necessários ao funcionamento do protótipo. A Figura 62 nos mostra um exemplo de requisição ao servidor, bem como retorno de objetos:

```
// Se estiver exibindo ofertas, cancela..
ib_ofertas_exibir      =      false;
// cria o objeto url
URL url = new URL(
    "http://" + IS_SERVIDOR + ":8084/TCommerceServer/retornaCategoria");

// cria o objeto httpurlconnection
URLConnection connection = (URLConnection) url.openConnection();
// seta o metodo
connection.setRequestProperty("Request-Method", "POST");
// seta a variavel para enviar parâmetros e ler o resultado
connection.setDoOutput(true);
connection.setDoInput(true);
connection.setRequestProperty("codigo_categoria", codigo_categoria);
// conecta com a url destino
connection.connect();
// Recebendo a resposta do servlet
io_lista_icones        =      new ArrayList<Icône>();
ObjectInputStream ois
    = new ObjectInputStream(connection.getInputStream());
//Recebe a lista e converte..
io_lista_icones        = (ArrayList<Icône>) ois.readObject();
ois.close();
// Agora chama o método para montar o painel de itens.
in_categoria_primeiro_item =      0;
in_categoria_ultimo_item   =      in_categoria_primeiro_item +
IN_ITENS_EXIBIDOS - 1;
montaPainelItens();
```

Figura 62. Requisição ao servidor e recebimento da lista de objetos serializados

A Figura 63 ilustra o tratamento desta mesma requisição no lado do servidor:

```

/**
 * Método que busca os ícones de itens de uma categoria e devolve ao cliente..
 */
public void retornaCategoria( HttpServletRequest request, HttpServletResponse response )
    throws ServletException, IOException
{
    try
    {
        // Objeto que realiza a interface com o banco de dados..
        ItemDao dao = new ItemDao();

        // Pesquisa os dados do código de categoria passada..
        String    codigo_categoria =    request.getHeader("codigo_categoria");

        // Busca os itens da categoria no banco..
        ArrayList<Icône> lista =    new ArrayList<Icône>();
        lista      =    dao.buscaListaCategoria( codigo_categoria,
                                                getServletContext().getRealPath("");

        // Carrega a imagens de cada um dos itens da lista..
        carregaImagem(lista);

        // Agora devolve a lista de objetos Icône ao cliente que fez a requisição..
        try
        {
            ObjectOutputStream ous = new ObjectOutputStream(response.getOutputStream());
            ous.writeObject(lista);
            ous.flush();
        }
        finally
        {
            System.out.println("retornaItem concluído..");
        }
    }
    catch(Exception e)
    {
        System.out.println("Exceção retornaItem: "+e.getLocalizedMessage());
        e.printStackTrace();
    }
}

```

Figura 63. Tratamento da requisição *retornaCategoria* pelo servidor, retornando objetos

A captura dos acionamentos de teclas do controle remoto é possibilitada pelo uso da implementação da interface ouvinte *UserInputEventListener* (pacote *com.sun.dtv.ui.event*) no desenvolvimento da classe principal (*Xlet*). Esta interface deve primeiro ser registrada pelo gerenciador de eventos de entrada do usuário (classe *UserInputEventManager*) associado ao dispositivo de exibição. Em seguida, sobrescreve-se seu método *userInputEventReceived()*, recebendo como parâmetro um objeto *UserInputEvent* que contém a origem do evento gerado – neste caso, um código de tecla acionada, ficando a implementação do método a critério do desenvolvedor. A Figura 64 ilustra o registro da interface e a sobrescrita do método ouvinte:

```

public class XletPrincipal extends Component
    implements Xlet, ScarceResourceListener, UserInputEventListener, FocusListener
{
    *** (continua) ***

    /* Uso do JavaDTV UI - Tratamento de eventos */
    UserInputEventManager manager =
        UserInputEventManager.getUserInputEventManager(currentScreen);
    RemoteControlEvent anyColoredKeyTyped =
        new RemoteControlEvent
            (
                null,
                java.awt.event.KeyEvent.KEY_TYPED,
                0, 0,
                RemoteControlEvent.VK_COLORED,
                KeyEvent.CHAR_UNDEFINED
            );

    try
    {
        anyColoredKeyTyped.reserve(true, -1, null);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

    manager.addUserInputEventListener(this, anyColoredKeyTyped);

    *** (continua) ***

    /**
     * Método para tratamento dos eventos do controle remoto
     */
    public void userInputEventReceived(UserInputEvent e)
    {
        com.sun.dtv.ui.event.KeyEvent event = (com.sun.dtv.ui.event.KeyEvent) e;
        if (event.getID() == KeyEvent.KEY_RELEASED)
        {
            return;
        }

        System.out.println(event.getKeyCode());

        // Chama o método que trata o evento e atualiza a tela..
        this.updateScene(event.getKeyCode());
    }
    *** (continua) ***
}

```

Figura 64. Implementação de *UserInputEventListener* e sobrescrita do ouvinte

Uma vez estabelecidos o ouvinte/tratador de eventos de usuário e o mecanismo de interatividade (acesso e requisições ao *web service*), o desenvolvimento da parte visual da interface - criação e disposição de painéis, ícones, conteúdo textual e demais elementos – é feito de forma semelhante àquele envolvendo componentes Swing, não havendo necessidade de pormenorizar essa etapa do desenvolvimento. A Figura 65 ilustra o protótipo exibindo os detalhes de um item selecionado pelo usuário:

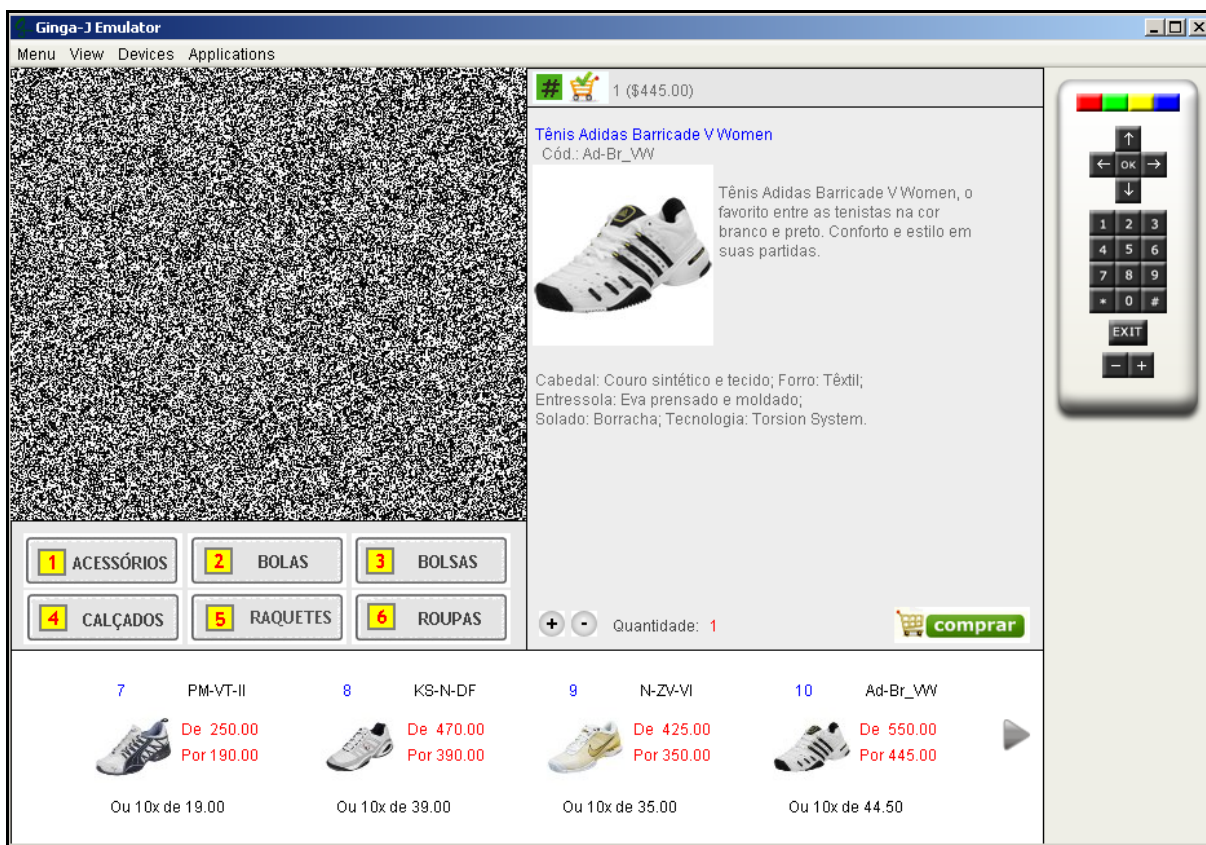


Figura 65. Exibição de *Calçados* no painel inferior, e detalhes de um item selecionado

Devido ao fato que nos casos de uso C05 (*login*) e C09 (efetuar compra) o protótipo envia e recebe do servidor remoto dados sigilosos via requisição - usuário, senha, número e código de segurança de cartão de crédito – considerou-se importante a implementação de um mecanismo de criptografia dessas informações antes de trafegarem pela rede. Optou-se pela utilização de uma chave criptográfica simétrica no padrão do algoritmo *Advanced Encryption Standard* (AES).

O algoritmo AES nasceu de um concurso promovido pelo governo dos Estados Unidos em 1997 com o objetivo de eleger um algoritmo substituto ao então padrão *Data Encryption Standard* (DES), que havia sido quebrado em um desafio via Internet. Após três anos e testes com 15 projetos candidatos, o algoritmo vencedor foi o *Rinjdael*, criado pelos belgas Vincent Rijmen e Joan Daemen, e escolhido por ser o melhor em quesitos como segurança, desempenho em hardware e software, entre outros (SOUZA; BORGES, 2006).

O AES suporta chaves com tamanhos variando em 128, 160, 192, 224 e 256 bits. Para utilização no protótipo, criou-se uma chave privada e simétrica de 256 bits, armazenada em um *array* de bytes. A mesma chave foi implantada no lado cliente e no lado servidor, sem o emprego de um mecanismo de chaves pública/privada, distribuição e registro de certificados de segurança. A classe utilitária *Cryptor* é responsável pela tarefa de geração do código secreto baseado na chave escolhida e de posse do código secreto, criptografar ou descriptografar (cifrar/decifrar) objetos String passados como parâmetro. Esta classe foi implementada com base no *framework Java Cryptography Extension 1.0 (JCE)*, que facilita a implantação de mecanismos de criptografia, geração e manipulação de chaves e assinaturas, entre outros. A Figura 66 exibe um trecho da classe, onde um parâmetro String é criptografado e devolvido:

```
// Atributos da classe..
private static final String METODO_ENCRIP_TACAO = "AES";
public static final byte[] CHAVE = { 117, 69, 115, 118, 63, -2, -80, 124,
                                     -26, 116, 72, -23, -54, 88, -7, 15};

/**
 * Método que criptografa uma string e devolve.
 */
public static String encriptar(byte[] key, String value)
    throws EncryptorException
{
    try
    {
        // Gera o código secreto baseado na chave definida..
        SecretKeySpec skeySpec = new SecretKeySpec(key, METODO_ENCRIP_TACAO);
        Cipher cipher = Cipher.getInstance(METODO_ENCRIP_TACAO);
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);

        // Criptografa o array e retorna como string..
        byte[] encrypted = cipher.doFinal(value.getBytes());
        return new BASE64Encoder().encode(encrypted);
    }
    catch (Exception e)
    {
        // Retorna a exceção..
        throw new EncryptorException("Erro ao criptografar informações "+ e.getMessage());
    }
}
```

Figura 66. Trecho da classe *Cryptor*, onde se vê a chave de 256 bits e a encriptação

Como nas duas pontas do sistema – cliente e servidor – são exigidas as tarefas de cifrar/decifrar, tanto o protótipo cliente como o *web service* utilizam os serviços da classe

Cryptor, inclusive com a mesma chave; logo a classe foi implantada nos dois ambientes, a exemplo das classes de objetos serializáveis representando as entidades envolvidos na interação (*Item*, *Icone*, *Cliente*, *ItemCompleto*, *ItemCarrinho* e *Transacao*).

Como se pode observar na Figura 66, a classe criptografa/descriptografa informações do tipo String. Sendo assim, foi necessária a criação de métodos que transformassem os objetos serializáveis em objetos String para que pudessem então ser criptografados. A classe *Cryptor* contém os métodos *public static byte[] getbytes(Object obj)* e *public static Object getObject(byte[] data)*. O primeiro transforma um objeto passado como parâmetro em um *array* de bytes e devolve. O segundo faz o caminho contrário. De posse de um *array* de bytes representando o objeto, facilmente instancia-se um objeto String a partir dos bytes, que será então submetida à criptografia. Por sua vez, uma String convertida em *array* de bytes, em seguida submetido ao método *getObject()* irá resultar em uma instância de *Object*, podendo finalmente transformar-se no objeto inicial, bastando apenas realizar a coerção, pois todos os objetos herdam de *Object*. A Figura 67 mostra o processo de cifragem no lado servidor e decifragem no cliente:

```

***** LADO SERVIDOR: TRANSFORMA O OBJETO EM STRING, ENCRIPTA E ENVIA COMO RESPOSTA *****
Cliente cliente          = dao.buscaCliente( cliente_login, cliente_senha);

// Transforma o objeto POJO Cliente em um array de bytes..
byte[] encoded          = Cryptor.getbytes(cliente);
// Encripta AES..
String objeto_criptografado = Cryptor.encriptar(Cryptor.CHAVE, new String(encoded));

// Envia a String..
ObjectOutputStream ous = new ObjectOutputStream(response.getOutputStream());
ous.writeObject(objeto_criptografado);
ous.flush();

*** LADO CLIENTE : RECEBE A STRING DA REQUISIÇÃO, DESCRIPTOGRAFA E TRANSFORMA EM OBJETO ***
Object objeto          = new Object();
ObjectInputStream ois  = new ObjectInputStream(connection.getInputStream());

// Recebe o objeto, descriptografa e transforma em objeto POJO Cliente
objeto                 = ois.readObject();
ois.close();

String objeto_criptografado = (String)objeto;
String objeto_descriptografado = Cryptor.decriptar(Cryptor.CHAVE, objeto_criptografado);
Cliente io_cliente = (Cliente) Cryptor.getObject(objeto_descriptografado.getBytes());

```

Figura 67. Cifragem/envio de objeto no servidor e recepção/decifragem no cliente

7.3 RESULTADOS OBTIDOS

O protótipo desenvolvido e testado com base no modelo iterativo incremental, embora tenham sido enfrentadas algumas dificuldades relacionadas aos componentes da interface gráfica, mostrou atender a todos os requisitos e casos de uso levantados durante a fase de análise e modelagem.

Diversos testes foram realizados, inclusive com a execução do *WebService* em uma máquina remota, onde vários ciclos foram completados com êxito em todas as suas etapas: seleção de categoria, exibição e seleção de itens, adição e exclusão de itens do carrinho, autenticação do usuário, finalização da compra, digitação dos dados do cartão e *logout* automático.

A Figura 68 exibe a tela de confirmação da compra:

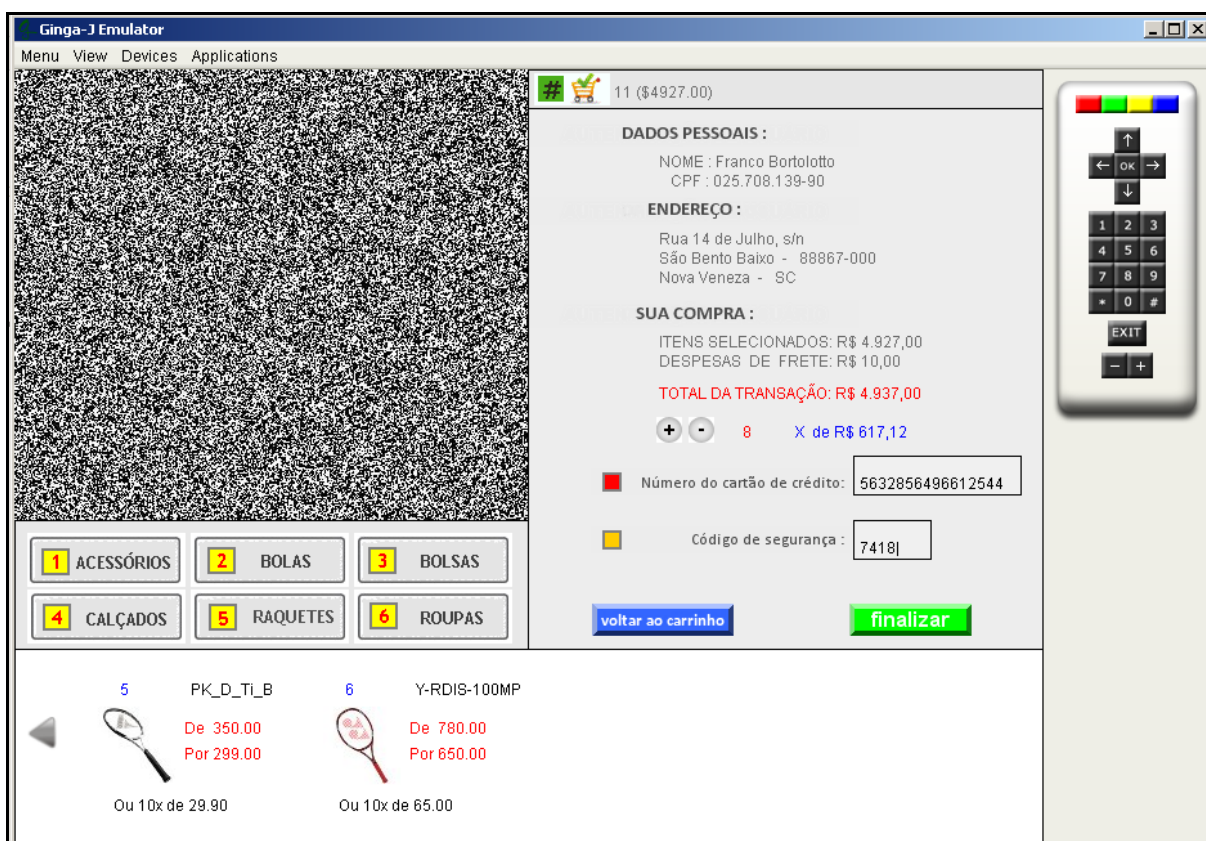


Figura 68. Tela de finalização/encerramento do processo de compra

Conforme explicado anteriormente, todos os componentes (*widgets*) de interface com o usuário foram construídos utilizando somente objetos etiquetas (classe *Label*), pois os demais *widgets* ainda estão indisponíveis. Esta foi a maior dificuldade encontrada no desenvolvimento, porque o componente *Label* não pode receber foco, entrada de texto e acionamento. Os componentes que mais fizeram falta no processo de desenvolvimento foram o botão (*Button*), campo de texto (*TextField*) e diálogo (*Dialog*), pois são adequados à tarefa de receber dados e eventos do usuário.

Para suprir essa deficiência de componentes, a utilização das teclas coloridas do controle remoto como gatilhos de eventos foi providencial, simulando o acionamento de botões ou opções nas cores correspondentes, tornando assim o processo de interação com o usuário o mais intuitiva possível. Na Figura 68, no painel que exibe os dados do cliente e da compra nota-se a presença de dois “botões” com opções: *Voltar ao carrinho*, na cor azul e *Finalizar*, na cor verde. Como se tratam apenas de *labels* representando botões, o acionamento da funcionalidade correspondente é feito clicando-se na tecla colorida do controle remoto correspondente à cor de cada botão – neste caso azul ou verde - e o evento correspondente é então disparado, como se o “botão” representado pela *label* tivesse recebido o foco e acionado realmente.

Pelo fato de não ter sido possível o desmembramento da interface em vários formulários, um para cada painel representado (todos os componentes estão contidos em um único *Form*), somado à falta do mecanismo de dupla bufferização – em fase de testes e implantação e ainda indisponível para uso – e à execução no emulador, a cada alteração ou atualização de componentes a tela da interface inteira “pisca”, causando incômodo ao utilizador. Entretanto, para fins de prototipagem, esta questão pode ser considerada relevante.

Na medida do possível o acionamento de opções foi facilitado, bastando pressionar o botão da cor correspondente. Entretanto em alguns casos, exigiu-se uma

combinação entre as teclas coloridas e outras teclas chave, como “#”, “OK” e códigos numéricos. Alguns exemplos são o acesso ao carrinho de compras, exclusão de itens do carrinho e seleção de categorias de produtos.

Observando-se as imagens capturadas da execução do protótipo, percebe-se o potencial do subsistema Ginga-J no que diz respeito à possibilidade de construção de interfaces ricas, elaboradas e com recursos avançados. Mesmo com todas as dificuldades encontradas em virtude das limitações decorrentes da indisponibilidade de praticamente todos os *widgets* possíveis, foi possível a construção de uma interface visualmente satisfatória e interativa, com navegação relativamente interativa. Isso mostra que uma vez estando todas as bibliotecas da especificação implementadas e em pleno funcionamento, os únicos limites para a construção de aplicações ricas, interativas e funcionais serão a criatividade do desenvolvedor e a capacidade do dispositivo onde o *Xlet* será executado.

No que diz respeito à questão da interatividade, mais uma vez os objetivos foram alcançados. O pacote *java.net* oferece acesso a mecanismos que implementam a comunicação via redes de forma relativamente simples, como *sockets*, conexões URL e datagramas, permitindo tarefas como requisições HTTP a servidores remotos, passagem e recebimento de parâmetros – inclusive objetos – autenticação e outros. Dessa forma, o canal de retorno/interatividade foi plenamente utilizado, sendo o funcionamento do protótipo – classificado como do tipo *Internet na TV* - totalmente dependente da interação com o servidor web remoto via rede, caracterizando assim a presença de uma interatividade do tipo remota, permanente (tempo real) e de nível 5 ao menos, considerando-se as classificações apresentadas no Capítulo 3.

Outros pacotes disponíveis na especificação Ginga-J, como *javax.microedition.io* e outros, oferecem a possibilidade de utilização de acesso a servidores web seguros - HTTPS – bem como aplicações envolvendo *sockets* seguros e certificados digitais.

Embora não faça parte dos objetivos do trabalho, a questão da segurança nas transações foi levada em conta, visto que a aplicação exige autenticação do usuário para que possa selecionar itens, e ao final do processo, envio de informações sigilosas como dados de cartão de crédito. Na autenticação por exemplo, a aplicação cliente envia ao servidor web uma requisição contendo os parâmetros *usuário* e *senha* em forma de string, parâmetros que se forem enviados como texto puro poderão ser facilmente interceptados por *sniffers* de rede, assim como o objeto *Cliente* enviado pelo servidor na resposta, contendo informações particulares – como CPF e endereço – do cliente cadastrado. Para oferecer um mínimo de sigilo na troca destas informações confidenciais, utilizou-se criptografia com chave simétrica AES, que embora utilize a mesma chave nas duas pontas, mostrou-se eficaz na tarefa de embaralhar os dados trafegados, tanto na requisição como na resposta do servidor, inclusive havendo criptografia de objetos serializados.

<pre> \$.x..E...Kp@... (Y...d.. .e.e.... \W...P..._... GET/TCommerceServer/autenticar HTTP/1.1.. Request- Method: POST.. usuário: 123456.. senha: 654321.. User-Agent: Java/1.6.0_17.. Host: 1 92.168.2.101:8084.. Accept: text/ html, image/gif, image/jpeg, *; q=.2, */*; q=.2. . Connection: keep-alive. ... </pre> <p style="text-align: center;">a)</p>	<pre>\$.x..E..]K,@... (U...d...e.b...0h...S.P...+I.. GET/TCommerceServer/autenticar HTTP/1.1.. Request-Method:POST.. dGTRfxRCP5i2+2eficNSiA==:lD53py5lplirw MyG+DzlfG==.. 5vTYncjh8p/SS7s9RLiFqQ==:YmGTtFiQ+0etk 35KEeTUVg==.. User-Agent:Java/1.6.0_17.. Host:192.168.2.101:8084.. Accept:text/html,image/gif, image/jpeg,*; q=.2,*/*;q=.2.. Connection:keep-alive.... </pre> <p style="text-align: center;">b)</p>
---	---

Figura 69. a) Envio de parâmetros sem criptografia; b) envio após criptografia

A Figura 69 ilustra a captura de pacotes durante a requisição à URL *TCommerceServer/autenticar*, onde os parâmetros usuário e senha são enviados ao servidor e um objeto *Cliente* é devolvido, ou *null* caso o cliente não esteja cadastrado. A captura foi feita utilizando-se a ferramenta analisadora de tráfego de rede *Wireshark*, com funcionalidade

idêntica à ferramenta *tcpdump* do Linux, mas em modo gráfico. Esta ferramenta permite capturar o tráfego dos pacotes que entram e saem por uma interface de rede, em vários protocolos.

Na Figura 69, em “a)” vê-se a captura dos pacotes da requisição em uma circunstância onde os parâmetros foram enviados de modo simples, no formato texto sem serem cifrados (criptografados), atentando-se para o modo como os valores tanto de chave como de valor ficam expostos de forma total. Já em “b)” pode-se observar a captura do tráfego da mesma requisição, mas agora as informações consideradas sigilosas – tanto os parâmetros quanto seus valores - foram criptografadas antes de serem enviadas pela requisição, sendo praticamente impossível para um eventual atacante identificar as informações enviadas, caso não conheça tanto o algoritmo criptográfico adotado quanto a chave utilizada.

Quanto à criptografia de objetos, passa a ser trivial a partir do instante em que estes possam ser representados em *array* de bytes e conseqüentemente, transformados em uma string que será por sua vez criptografada. Na outra ponta, basta percorrer o caminho inverso, e ao final ter-se-á o objeto original. A Figura 70 ilustra em um *trace* (*System.out.println*) o objeto Cliente, enviado em resposta à requisição exibida na Figura 69, após ser submetido à criptografia e pronto para ser enviado ao cliente:

```
Request cliente:123456
Request senha:654321
path - C:\TCommerceServer\build\web
String select: select * from CLIENTE where codigo_login = ? and codigo_senha = ?
buscaCliente- cliente: Franco
- Array autenticar: [B@1f6f27b
- Objeto criptografado:
AkDIO/GQ2BQko0iQR1BHaKB/TgXABC6nHrXjNfjQng86UMp48oJQBRxpo8ikuj3czQERQxhIC8c5
J7XJ3IQLhUU7ACMwLLch+hd1kj2eq+FX+cAlj5iPkYv0zvOQHHP6/3Lg2174eCMZcQTm61JrGzrT
3RK1RY5x60Khs7Emp9QZ+tZZ6e8SLBbTy4Zye7vfyBBz+4PpDtCSpQ0THX719sgpHnbzcBg3Oc3a
vFwZLrC179B/JTb57rUTO2LZJS2BuvEP+cR/U11K51iZ+wpfHKQwE1VmgIf+rxq2XbV9AHG5YYG
q2iEYeDo3TrcaTAdZG7E1mzjvRap/W/YHeN1tOXavWx4rIKZac1NKOyxe6hy/fKzR3QH+axruvt4
6zseWN4KR/cQo7S3iTv+5aTui3QTjwVHw79QW5fCON3IzmiSBqaZPZ7q5sjL8J65Ho2G+EEoHLZZ
sy3q1Abnyd+hLMPvXdG/hSMzpv1QOHYK4SjiqTYj5YnWQX5R*WKZvVxGOH41qIoD6M4YR5X9LbkX
mUvKUE1gvCTdzQzc5LF6EHLc9I15sxupqV9VcpVP14hqumY1EtUnypb/Yf+Jc7/icCdFcciy1JEG
TW11CsVcM0+/IKfIh2Rs6gyEFHlOC4NifLCq0lmSwRdanxbXmcVLPg==
retornaCliente concluído..
```

Figura 70. Objeto *Cliente* cifrado no *WebService*, antes de ser enviado ao cliente

CONCLUSÃO

O protótipo desenvolvido – ápice dos trabalhos realizados durante a pesquisa – serve como demonstrativo prático das múltiplas possibilidades decorrentes da utilização da especificação Ginga no desenvolvimento de aplicações interativas para a TV Digital.

Em particular as bibliotecas do subsistema procedural Ginga-J, objeto específico deste estudo, que apesar das dificuldades enfrentadas e limitações encontradas- em virtude de que boa parte da especificação ainda esteja em trabalho de implantação ou implementação – possibilitou o desenvolvimento de um protótipo com interface gráfica consideravelmente rica e funcional, com interatividade local satisfatória e na medida do possível intuitiva.

No tocante à questão da interatividade, os recursos de rede oferecidos pela especificação viabilizaram o estabelecimento de um canal de interatividade remota e dedicada via canal de retorno (Internet) de maneira simples, tal qual em aplicações *web*.

Vale frisar que o levantamento bibliográfico e estudo realizado com base no tema Televisão Digital Interativa, iniciando pela visão genérica e culminando nos pontos específicos (o *middleware* Ginga e seu subsistema procedural), constituiu-se como etapa fundamental do trabalho. Um bom estudo da estrutura e funcionamento dos módulos componentes do ambiente, do mecanismo de interatividade na TV Digital – sua complexidade, impacto social e aspectos tecnológicos – e do histórico dos três sistemas pioneiros abriu caminho para uma adequada compreensão da estrutura do Sistema Brasileiro de Televisão Digital e seus módulos, tanto do ponto de vista tecnológico como no social.

O padrão nipo-brasileiro é considerado como vanguarda atualmente, e vem sendo adotado por outros países latino-americanos, graças ao fato de que além de reunir o que de melhor se fez nos sistemas pioneiros, apresenta inovações como padrões de transmissão e

codec mais robustas e eficientes, possibilidade de recepção móvel e digital e um middleware voltado à atender as necessidades de inclusão social e digital.

No tocante à questão da interatividade, é evidente a necessidade de se estabelecer políticas que permitam a criação de uma estrutura de comunicações que alcance as classes mais baixas da sociedade – justamente aquelas que são alvo dos objetivos primordiais estipulados pelo Governo Federal quando da implantação do SBTVD: promover a inclusão social através da inclusão digital de forma gratuita a toda população. Sem o emprego de um canal de retorno via Internet, as possibilidades de interatividade na TV Digital são limitadas; logo, a questão de se oferecer acesso de qualidade e baixo custo à população carente precisa ser acompanhada, não só pelo setor público, mas pela sociedade brasileira em um conjunto de esforços.

Os objetivos – tanto geral como específicos – foram alcançados com sucesso e devidamente documentados nesta pesquisa, servindo como para trabalhos futuros, na medida do possível. Embora não tenha sido estipulado como objetivo neste trabalho, considerou-se importante a implantação de um mecanismo criptográfico proporcionando um mínimo de segurança nas transações, etapa esta concluída igualmente com relativo sucesso.

Como sugestões de trabalhos futuros, entre outras estão:

- a) desenvolvimento de aplicações com interface mais rica, lançando mão dos componentes *lwuit* que estão em fase de implementação pelo Ginga CDN;
- b) emprego do mecanismo de ponte no desenvolvimento de aplicações híbridas, envolvendo os dois paradigmas – declarativo (NCL) e procedural (Java);
- c) aplicações Ginga-J envolvendo múltiplos dispositivos de interação;
- d) emprego de dispositivos avançados, como *smart card* e leitores biométricos;
- e) utilização de mecanismo de criptografia com distribuição de chaves públicas e certificados de autenticidade.

REFERÊNCIAS

ABNT. Associação Brasileira de Normas Técnicas. **NBR 15606-4:2010**: Ginga-J – Ambiente para a execução de aplicações procedurais. 91 f. Primeira edição (válida a partir de 13.05.2010), Brasília, 2010. Disponível em: <http://www.dtv.org.br/download/pt-br/ABNTNBR15606-4_2010Ed1.pdf>. Acesso em 10 mai. 2010.

_____. Associação Brasileira de Normas Técnicas. **PROJETO 00:001.85-006/4**: Ginga-J: Ambiente para a execução de aplicações procedurais. 36 f. Primeiro projeto, Brasília, set. 2007. Disponível em: <<http://lince.dc.ufscar.br/home/pos-graduacao/Material%20TVD/parte4-gingaJ.pdf>>. Acesso em 15 out. 2009.

ALMEIDA JUNIOR, Edson. TV Digital brasileira: O que é? O que vai mudar? **Revista Científica Tecnólogos**, Recife, ed. n. 3, set. 2008. Disponível em: <http://www.unibratrec.com.br/revistacientifica/n3_artigos/n3_almeida_jr.pdf>. Acesso em: 15 out. 2009.

ALVES, Luiz Gustavo Pacola. **Collaboratvware**: uma infra-estrutura ciente de contexto para suporte a participação colaborativa no cenário da TV Digital interativa. 2008. 203 f. Dissertação (Mestrado em Engenharia). Área de concentração: Sistemas Digitais – USP: Escola Politécnica, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-20072009-114734/>>. Acesso em: 5 out. 2009.

AMATO NETO, João. Análise da emergência da tv digital e seus impactos na cadeia produtiva eletroeletrônica brasileira. **Journal of Technology Management & Innovation**, Santiago, v.1, n.5, p. 53-68, dec. 2006. Disponível em: <<http://www.jotmi.org/index.php/GT/article/view/art30>>. Acesso em: 5 out. 2009.

ANDREATA, Jomar Alberto. **InteraTV**: um portal para aplicações corporativas em TV Digital interativa utilizando a plataforma MHP. 2006. 110 f. Dissertação (Mestrado em Engenharia Elétrica). Área de concentração: Automação e Sistemas – Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <<http://www.das.ufsc.br/~montez/publications/2006%20Jomar.Andreata.pdf>>. Acesso em: 5 out. 2009.

ARAUJO, Rodrigo Cascão. TV Digital interativa: uma abordagem dos novos desafios para as instituições de PD&I. IN: II Seminário sobre a organização da Informação Digital, 2009, Campinas. **Anais...** Campinas: EMBRAPA, 2009. Disponível em: <http://www.cnptia.embrapa.br/infodigital/palestras/rodrigo_interatividade.pdf>. Acesso em: 10 mar. 2010.

ARNOLDO, Marcos Vinicius Henke. Ferramentas de acessibilidade para tv digital interativa com Java. In: Seminário de Informática - RS (SEMINFO RS'2008), 2008, Torres. **Anais eletrônicos...** Torres: ULBRA, 2008. Disponível em: <<http://www.seminfo.com.br/anais/2008/pdfs/tcc/30-50880.pdf>>. Acesso em: 5 out. 2009.

AYRES, Marcelo. TV Digital inaugura patamar de interatividade e integração. **Portal UOL Tecnologia**, São Paulo, 14 jun. 2007. Disponível em: <<http://tecnologia.uol.com.br/produtos/ultnot/2007/06/14/ult2880u373.jhtm>> . Acesso em: 03 mar. 2010.

BARBARA, Felipe Nogueira; SOARES, Luiz Fernando Gomes. Desenvolvimento de aplicações declarativas e testes de conformidade para a TV Digital brasileira. IN: XVI Seminário de Iniciação Científica da PUC-Rio, 2008, Rio de Janeiro. **Anais...** Rio de Janeiro: PUC, 2008. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2008/relatorios/ctc/inf/inf_marcoantonio.pdf>. Acesso em: 02 mar. 2010.

BASTOS, Pollyanna. A TV Digital e as Novas Possibilidades de Interatividade. IN: XXXI Congresso Brasileiro de Ciências da Comunicação, Natal, 2008. **Anais...** Natal: Intercom, 2008. Disponível em: <<http://www.intercom.org.br/papers/nacionais/2008/resumos/R3-0768-1.pdf>>. Acesso em: 15 out. 2009.

BECKER, Valdecir et al. Datacasting e Desenvolvimento de Serviços e Aplicações para TV Digital Interativa. In: TEIXEIRA, Cesar Augusto Camilo; BARRÉRE, Eduardo; ABRÃO, Iran Calixto. (Org.). **Web e Multimídia: Desafios e Soluções**. Poços de Caldas, 2005, v. 01, p. 01-30. Disponível em: <<http://www.itvproducoesinterativas.com.br/pdfs/A-Datacasting-webmidia.pdf>>. Acesso em 10 out. 2009.

BECKER, Valdecir; MONTEZ, Carlos. TV Digital Interativa: conceitos e tecnologias. IN: Minicursos WebMedia & LA-Web, 2004, Ribeirão Preto. **Anais...** Ribeirão Preto: WebMedia, 2004. Disponível em: <<http://www.itvproducoesinterativas.com.br/pdfs/A-TV-Digital-Interativa-Conceitos-Tecnologias.pdf>>. Acesso em: 05 mar. 2010.

BEDICKS JUNIOR, Gunnar. **Sintonizador-demodulador para o Sistema Brasileiro de TV Digital**. 2008. 169 f. Tese (Doutorado em Engenharia Elétrica). Área de concentração: Sistemas Eletrônicos – Universidade de São Paulo: Escola Politécnica, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-26092008-101922/>>. Acesso em: 5 out. 2009.

BLOG. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Blog&oldid=22400641>>. Acesso em: 2 nov. 2010.

BOQUIMPANI, Aguinaldo. Explorando o Ginga-J: como atingir cenários reais de interatividade na TV. IN: JustJava, 2009, São Paulo. **Anais...** São Paulo:SUCESU-SP, 2009. Disponível em:
<<http://www.sucesusp.org.br/mailling2009/congresso/justjava/apresentacoes/AuditorioPrincipal/TVDigital.pdf>>. Acesso em: 2 nov. 2009.

BRACKMANN, Christian Puhmann. **Sistema brasileiro de TV Digital**. 2008. 57 f. Trabalho Individual do Programa de Pós-Graduação em Informática, Centro Politécnico, Universidade Católica de Pelotas, Pelotas, 2008. Disponível em:
<http://www.tvdi.inf.br/upload/artigos/sbtvd_-_ti1.pdf>. Acesso em: 05 out. 2009.

BRACKMANN, Christian P.; SILVA, Cauane B.; LUZZARDI, Paulo Roberto G.. TVMARK: SET-TOP BOX BENCHMARKING. IN: IV CONGRESSO SUL CATARINENSE DE COMPUTAÇÃO, 2008, Criciúma. **Anais...** Criciúma: UNESC, 2008. Disponível em: <<http://www.infociencia.info/artigos/ARTr544.pdf>>. Acesso em: 6 out. 2009.

BRASIL. **Decreto nº 4901, de 26 de novembro de 2003**. Disponível em:
< <http://www.mc.gov.br/tv-digital/decreto-no-4901-de-26-de-novembro-de-2003/>> Acesso em 15 out. 2009.

_____. **Decreto nº 5820, de 29 de junho de 2006**. Disponível em:
< http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Decreto/D5820.htm> Acesso em 15 out. 2009.

BREUNIG, Mônica Caroline; MARQUES, Viviane. A digitalização da TV. IN: XXXI Congresso Brasileiro de Ciências da Comunicação – Natal. **Anais...** Natal, 2008. Disponível em:
<<http://www.intercom.org.br/papers/nacionais/2008/resumos/R3-1569-1.pdf>>. Acesso em: 15 out. 2009.

BROADCAST. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em:
<<http://pt.wikipedia.org/w/index.php?title=Broadcast&oldid=22385976>>. Acesso em: 2 nov. 2010.

BURGER, Roland A. et al. A Survey of Digital TV Standards China. IN: 2nd International Conference on Communications and Networking in China (ChinaCom), Shanghai, 2007. **Anais...** Shanghai: IEEE Press, 2007, pp. 687-696. Disponível em: < <http://www.tvdi.inf.br/upload/artigos/china-dtv-a4.pdf>>. Acesso em: 10 mar. 2010.

CAPDA – Comitê das Atividades de Pesquisa e Desenvolvimento na Amazônia. **TV Digital interativa**: programa prioritário do Subgrupo de Trabalho 2. Manaus, 2004. Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus – CT-PIM. Disponível em: <http://www.ctpim.org.br/tv_digital.pdf>. Acesso em: 05 mar. 2010.

CARVALHO, Eduardo Rodrigues. **Uma plataforma modular para testes com interatividade na TV Digital brasileira**. 2008. 119 f. Dissertação (Mestrado em Engenharia). Área de concentração: Sistemas Eletrônicos – USP: Escola Politécnica, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-30052008-110812/>>. Acesso em: 5 out. 2009.

CARVALHO, Rafael. Como estruturar seu ambiente de desenvolvimento para o Ginga-NCL. **Ginga DF: TV Digital se faz com Ginga!** Seção Blog Ginga, 07 mai. 2010. Disponível em: <<http://www.gingadf.com/blogGinga/?cat=12>>. Acesso em: 20 mai. 2010.

CASTRO, Jobert Marssola; CODATO, Henrique. Sistema Brasileiro de Televisão Digital: marketing, tendências e aplicabilidade do audiovisual. IN: 7º Encontro Internacional de Arte e Tecnologia, 2008, Brasília. **Anais...** Brasília: #7 ART, 2008 Disponível em: <http://www.jobertcastro.com/artigo_sbtvd.pdf>. Acesso em: 10 out. 2009.

CICLO de vida interativo e incremental. IN: **Página Wiki do Instituto Federal de Santa Catarina – IF-SC**. Última edição em 20 out. 2006. Disponível em: <http://wiki.sj.cefetsc.edu.br/wiki/index.php/Ciclo_de_Vida_Iterativo_e_Incremental>. Acesso em: 10 ago. 2010.

CNN and Echo Star launch interactive TV news service. **Informity – Connected Vision**. Section News, 21 feb. 2006. Disponível em: <<http://informity.com/news/2006/02/21/cnnandechoStar/>>. Acesso em: 22 mai. 2010.

CODEC. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Codec&oldid=21690052>>. Acesso em: 2 nov. 2010.

CRUZ, Daniel Alves. **O que é a TV Digital**. Home-page da ABERT – Associação Brasileira de Rádio e Televisão – seção artigos, [2009]. Disponível em: <<http://www.abert.org.br/novosite/biblioteca/TVDigital.pdf>>. Acesso em 10 out. 2009.

CRUZ, Vitor Medina; MORENO, Marcio Ferreira; SOARES, Luiz Fernando Gomes. **TV Digital para dispositivos portáteis - middlewares**. 2008. 74 f. Monografia (Ciência da Computação) - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/08_03_cruz.pdf>. Acesso em: 15 out. 2009.

DALL'ANTONIA, Juliano Castilho. **Cadeia de Valor FUNTTEL – Projeto Sistema Brasileiro de TV Digital**. OS 40539. Versão PD 30.12.36ª.002A/RT-02-AB, 95 f. Campinas, CPqD, 2004. (Relatório Técnico). Disponível em: http://sbtvd.cpqd.com.br/cmp_tvdigital/divulgacao/anexos/26_42_cadeia_de_valo.pdf. Acesso em 16 out. 2009.

DAMASCENO, Jean Ribeiro. **Middleware Ginga**. Artigo apresentado à disciplina de Fundamentos de Sistemas Multimídia, Escola de Engenharia, Universidade Federal Fluminense, Niterói, 2008. Disponível em: <<http://www.midiacom.uff.br/~debora/fsmm/trab-2008-2/middleware.pdf>>. Acesso em: 30 out. 2009.

DARÓS, Joel Alexandre. **Interfaces de desenvolvimento de aplicações para TV Digital baseado no middleware MHP**. 2004. 53 f. Trabalho de Conclusão de Curso (Bacharelado) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2004. Disponível em: <<http://campeche.inf.furb.br/tccs/2004-II/2004-2joeladarosap.pdf>>. Acesso em: 18 out. 2009.

DE LUCA, Cristina. Você já ouviu falar do Ginga CDN? **Convergência Digital**. Coluna Circuito, 25 fev. 2009. Disponível em: <<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=17844&sid=54>>. Acesso em: 15 mai. 2010.

_____. TV Digital: Universidade da Paraíba libera beta do OpenGinga. **Convergência Digital**. Coluna Circuito, 02 jul. 2009. Disponível em: <<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=19437&sid=54>>. Acesso em: 10 abr. 2010.

DUARTE, Diogo Alano. **Desenvolvimento de um protótipo de interatividade para TV Digital através do middleware Ginga**. 2009. 61 f. Trabalho de Conclusão de Curso (Bacharelado) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2009.

ENTENDENDO o ambiente OpenGinga. **Google Grupos: Desenvolvimento em Ginga Java para TV Digital**, out. 2008. Disponível em: <http://groups.google.com.br/group/gingajava/browse_thread/thread/ac6ff0e838c2470/1e0a8b4cf61866d1?lnk=gst&q=openginga#1e0a8b4cf61866d1>. Acesso em: 05 mai. 2010.

ETCHARTE, Valério Pereira. **Receptores e canal de interatividade para o Sistema Brasileiro de TV Digital**. 2008. Artigo apresentado à disciplina de Fundamentos de Sistemas Multimídia, Escola de Engenharia, Universidade Federal Fluminense, Niterói, 2008. Disponível em: <<http://www.midiacom.uff.br/~debora/fsmm/trab-2008-2/receptor.pdf>>. Acesso em: 06 mar. 2010.

FERREIRA, Simone de Lucena. **Possibilidades para a educação em rede com a TV Digital no Brasil**. 2008. 233 f. Tese (Doutorado em Educação). Área de concentração: Educação – Universidade Federal da Bahia: Faculdade de Educação, Salvador. Disponível em: <http://www.dominiopublico.gov.br/pesquisa/DetalheObraForm.do?select_action=&co_obra=137220>. Acesso em: 5 out. 2009.

FLOG. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Flog&oldid=22401300>>. Acesso em: 2 nov. 2010.

GINGACDN. **GINGA Code Development Network**. Página de acompanhamento do projeto Ginga CDN, 2009. Disponível em: <<http://gingacdn.lavid.ufpb.br/>>. Acesso em: 01 jun. 2010.

GINGA Digital TV Middleware Specification. **TV Interativa se faz com Ginga**. Página de apresentação do middleware Ginga, 2008. Disponível em: <<http://ginga.org.br/>>. Acesso em: 01 out. 2009.

GINGA-NCL. **Wiki dot Ginga RN – TV Digital interativa**. Seção Ginga-NCL, última edição em 11 mai. 2009. Disponível em: <<http://gingarn.wikidot.com/gingancl>>. Acesso em: 05 mai. 2010.

GUIMARÃES, Rodrigo Laiola; COSTA, Romualdo M. de Resende. **Interatividade e sincronismo em TV Digital**. 2006. 64 f. Monografia apresentada como requisito parcial para aprovação na disciplina de Tópicos de Hipertexto e Multimídia II, Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006. Disponível em: <http://homepages.cwi.nl/~guimarae/academic/laiola_monografia_interatividade.pdf>. Acesso em: 10 out. 2009.

IDEC – Instituto Brasileiro de Defesa do Consumidor. **TV Digital no Brasil: para não ficar fora do ar**. Junho, 2006. Disponível em: <http://www.idec.org.br/arquivos/TV_digital.pdf>. Acesso em: 11 mar. 2010.

HABBEMA, Hugo. Modelos prescritivos de desenvolvimento de software. **Habbema's Blog**. Postado em 27 ago. 2009. Disponível em: <<http://hugohabbema.blogspot.com/2009/08/modelos-prescritivos-de-desenvolvimento.html>>. Acesso em: 15 ago. 2010.

INTEL e Sun fecham acordo para o Ginga-J. **e-Thesis – Tecnologia e negócios**. 2009. Disponível em: <http://www.e-thesis.inf.br/index2.php?option=com_content&do_pdf=1&id=4890>. Acesso em: 4 nov. 2009.

INTERVOZES. **TV Digital: Princípios e Propostas para uma Transição Baseada no Interesse Público**. Janeiro, 2006: Disponível em: <<http://www.abranet.org.br/doc/tvdigital-intervozes.pdf>>. Acesso em: 11 nov. 2010.

JAVA DTV para Ginga-J sem royalties. **Produção Profissional**, São Paulo, v. 1, n.85, jan. 2009. Disponível em: <<http://www.producaoprofissional.com.br/files/editions/ed1236005099.pdf>>. Acesso em: 4 out. 2009.

KNEIPP, Valquiria Passos. Trajetória da TV Digital no Ceará e a interatividade do Ginga. IN: 1º Simpósio Internacional de Televisão Digital (SIMTVD), 2009, Bauru. **Anais...** Bauru: SIMTVD, 2009. Disponível em: <[http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/KNEIPP%20-%20Trajet%20da%20TV%20Digital%20no%20Cear%e1%20\(500-511\).pdf](http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/KNEIPP%20-%20Trajet%20da%20TV%20Digital%20no%20Cear%e1%20(500-511).pdf)>. Acesso em: 02 mar. 2010.

KULEZSA, Raoni. **API JavaTV: criando e controlando aplicações Java no Ginga**. Overmedia Networks – Treinamento Ginga Módulo 1, 2009. Disponível em: <<http://graphs.ucpel.tche.br/tvdi/OficinaGingaJ-Parte2.pdf>>. Acesso em 02 mar. 2010.

KULEZSA, Raoni; FERREIRA, Jefferson. Desenvolvimento Ginga-J - Java DTV- OpenGinga. IN: Fórum Internacional de Software Livre: Oficina Ginga-J, 2009, Porto Alegre. **Anais...** Porto Alegre: FISL, 2009. Disponível em: <<http://graphs.ucpel.tche.br/tvdi/OficinaGingaJ-Parte1.pdf>>. Acesso em: 15 out. 2009.

KUNG, Fabio Correia; KENZO, Marcelo Okamura; ZANINI, Rafael Anicet. **USBTv: a TV Digital no seu computador**. 2007. 96 f. Dissertação (Bacharelado em Engenharia). Área de concentração: Engenharia da Computação – USP: Escola Politécnica, São Paulo. Disponível em: <<http://www.pcs.usp.br/~pcspf/2007/Cooperativo%202007/PCS%202050%20COOP%20Grupo%20%2802%29/grupo2c.pdf>>. Acesso em: 7 out. 2009.

LAGALLY, Michael; PÄTZOLD, Jens; SUN Microsystems. New tv standard for digital tv in Brazil. IN: Java Mobile, Media & Embedded Developer Days Conference, 2009, Santa Clara **Anais...** Santa Clara, 2009. Disponível em: <http://www.tvdi.inf.br/upload/artigos/new_tv_standard_for_digital_tv_in_brazil.pdf>. Acesso em: 4 nov. 2009.

LAMAS, Amilton da Costa et al. Canal de interatividade em TV Digital. **Caderno CPqD Tecnologia**, Campinas, v.1, n.1, p. 29-36, jan./dez. 2005. Disponível em: <http://www.cpqd.com.br/cadernosdetecnologia/Vol1_N1_jan_dez_2005/pdf/artigo2_Lamas.pdf>. Acesso em: 10 mar. 2010.

LEMOS, André. Anjos interativos e retribalização do mundo: sobre interatividade e interfaces digitais. **Tendências XXI**, Lisboa, n. 2, p. 19-29, set. 1997. Disponível em: <<http://www.facom.ufba.br/ciberpesquisa/lemos/interac.html>>. Acesso em: 15 mar. 2010.

LIMA, Felipe Martins. **Protocolo de aplicação para jogos de tabuleiro para ambiente de TV Digital**. 2007. Artigo apresentado à disciplina de Fundamentos de Sistemas Multimídia, Escola de Engenharia, Universidade Federal Fluminense, Niterói, 2007. Disponível em: <<http://www.midiacom.uff.br/~debora/fsmm/trab-2007-2/protocolo.pdf>>. Acesso em: 06 mar. 2010

LIMA JUNIOR, Walter Teixeira; GARRIDO, Lina. Interatividade na TV Digital: a interface como metáfora da arquitetura computacional. IN: 1º Simpósio Internacional de Televisão Digital (SIMTVD), 2009, Bauru. **Anais...** Bauru: SIMTVD, 2009. Disponível em: <[http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/LIMA%20JUNIOR%3b%20GARRIDO%20-%20Interatividade%20na%20TV%20Digital%20\(552-569\).pdf](http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/LIMA%20JUNIOR%3b%20GARRIDO%20-%20Interatividade%20na%20TV%20Digital%20(552-569).pdf)>. Acesso em: 02 mar. 2010.

LINGUAGEM de script. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Linguagem_de_script&oldid=22356749>. Acesso em: 2 nov. 2010.

LOPES, Denise Maria Moura da Silva. Sistema Brasileiro de TV Digital: caminhos percorridos e implantação. IN: V Congresso Nacional de História na Mídia, 2007, São Paulo. **Anais...** São Paulo: Intercom, 2007. Disponível em: <<http://www.rp-bahia.com.br/biblioteca/hist-midia2005/resumos/R0097-1.pdf>>. Acesso em: 09 out. 2009.

LOWTRUX photostream. TCommerce for online tv app. **Flickr from Yahoo!** Taken on 20 nov. 2009. Disponível em: <<http://www.flickr.com/photos/lowtrux/4119091121/>>. Acesso em: 22 mai. 2010.

MACIEL, Danilo Leal; SILVA JUNIOR, Luis; SOUZA, Cidcley Teixeira. MulTV: uma ferramenta de autoria de aplicações interativas para o Sistema Brasileiro de TV Digital. IN: 35ª InfoBrasil TI & Telecom e II Congresso Tecnológico InfoBrasil, 2009, Fortaleza. **Anais...** Fortaleza: InfoBrasil, 2009. Disponível em: <<http://www.infobrasil.inf.br/iConstructor/Custom/anais2009/MulTV%20—20Uma%20Ferramenta%20de%20Autoria%20de%20Aplica%C3%A7%C3%B5es%20Interativas%20para%20o%20Sistema%20Brasileiro%20de%20TV%20Digit.pdf>>. Acesso em: 12 out. 2009.

MACKLIN, Ben. What every marketer needs to know about iTV. **eMarketer™**. New York, 2002. Disponível em: <http://banners.noticiasdot.com/termometro/boletines/docs/consultoras/emarketer/2002/emarketer_itv.pdf>. Acesso em: 10 abr. 2010.

MARGALHO, Mauro; FRANCÊS, Carlos Renato; COSTA, João Crisóstomo Weyl. Canal de retorno para TV Digital com interatividade condicionada por mecanismo de sinalização contínua e provisionamento de banda orientada a QoS. **IEEE Latin America Transactions**, São Paulo, vol. 5, n.5, p. 367-372, set. 2007. Disponível em: <http://ewh.ieee.org/reg/9/etrans/ieee/issues/vol5/vol5issue5Sept.2007/5TLA5_13Margalho.pdf>. Acesso em: 6 out. 2009.

MASCARENHAS, Alan; NORONHA, Karla; TAVARES, Olga. TV Digital: o *middleware* Ginga e a interatividade brasileira. IN: XXXII Congresso Brasileiro de Ciências da Comunicação, Curitiba, 2009. **Anais...** Curitiba: Intercom, 2009.

Disponível em:

<<http://www.intercom.org.br/papers/nacionais/2009/resumos/R4-2361-1.pdf>>. Acesso em: 15 out. 2009.

MATOS, Felipe Stutz; JULIÃO, Wilhelm Santanna; SANTOS, Neide. TV Digital e educação à distância: discutindo alguns pontos e propondo um protótipo de sistema. IN: XVIII Simpósio Brasileiro de Informática na Educação (SBIE), 2007, São Paulo. **Anais...** São Paulo: Mackenzie, 2007.

Disponível em: <<http://200.169.53.89/download/CD%20congressos/2007/SBIE2007/fscommand/Full/33826.pdf>>. Acesso em: 02 mar. 2010.

MÉDOLA, Ana Silvia Lopes Davi; TEIXEIRA, Lauro Henrique de Paiva. Aspectos da TV Digital interativa: como pode ficar a nova televisão do ponto de vista do usuário. IN: Fórum da Diversidade e Igualdade: cultura, educação e mídia – Bauru, 2007. **Anais...** Bauru: UNESP, 2007. Disponível em:

<<http://www.faac.unesp.br/publicacoes/anais-comunicacao/textos/35.pdf>>. Acesso em: 10 out. 2009.

MONTEIRO, Bruno de Souza. **Amadeus-TV: portal educacional na TV Digital integrado a um sistema de gestão de aprendizado**. 2009. 116 f. Dissertação (Mestrado em Ciência da Computação)– Universidade Federal de Pernambuco: Centro de Informática, Recife.

Disponível em: <http://www.cin.ufpe.br/~ccte/publicacoes/2009_Dissertacao_BrunoMonteiro.pdf>. Acesso em: 02 mar. 2010.

MONTEIRO, Marcelo Souto Maior. **TV interativa e seus caminhos**. 2002. 84 f. Dissertação (Mestrado em Computação). Área de concentração: Engenharia de Computação – Instituto de Computação, Universidade Estadual de Campinas, Campinas. Disponível em:

<<http://libdigi.unicamp.br/document/?code=vtls000294986>> Acesso em: 10 abr. 2010.

MONTEZ, Carlos; BECKER, Valdecir. **TV Digital interativa: conceitos, desafios e perspectivas para o Brasil**. 2. ed. ampliada e revisada. Florianópolis: da UFSC, 2005. 201 p. Disponível em: <http://www.labtvd.com.br/novo/attachments/061_TV-Digital-Interativa_2a_EDICAO.pdf>. Acesso em: 10 mar. 2010.

NERY E SILVA, Lincoln David et al. Suporte para desenvolvimento de aplicações multiusuário e multidispositivo para tv digital com Ginga. **Revista T&C Amazônia**, Manaus, ano V, n.12, p. 75-84, out 2007. Disponível em:
<https://portal.fucapi.br/tec/imagens/revistas/011_ed012_suport_desen_aplic_multusua_dispo_TVdigi_Ginga.pdf>.
Acesso em: 12 out. 2009.

OLIVEIRA, Amanda. **TV Digital brasileira: momento de transição e adaptação: em foco o telejornalismo**. 2008. 80 f. Monografia Curso (Bacharelado) - Curso de Comunicação Social - Jornalismo, Departamento de Artes e Humanidades, Universidade Federal de Viçosa, Viçosa, 2008. Disponível em:<http://www.com.ufv.br/pdfs/tccs/2008/2008_amandadeoliveira_tvdigitalbrasileira.pdf>.
Acesso em: 02 mar. 2010.

OLIVEIRA, Bruno Julian Dias. **Um estudo de caso entre o Ginga-J e o Ginga-NCL no âmbito de aplicações interativas residentes**. 2010. 47 f. Monografia (Bacharelado em Ciência da Computação). Área de concentração: Interação humano-computador. Universidade Federal da Paraíba, João Pessoa, 2010. Disponível em:
<http://groups.google.com/group/devdtv/attach/f0c5954fadbf7742/Monografia_BrunoDias+v.2.0.pdf?part=4>.
Acesso em: 10 mai. 2010.

OPENGINGA. **OpenGinga** - Implementação de referência do middleware brasileiro de TV Digital. Disponível em: <<http://www.openginga.org/>>. Disponível em cache:
<<http://webcache.googleusercontent.com/search?q=cache:S8i2buSmMPgJ:www.openginga.org/about.html+O+que+%C3%A9+OpenGinga+%3F&cd=1&hl=pt-BR&ct=clnk&gl=br>>
Acesso em: 01 mar. 2010

OPENGINGA LIBERADO. **Ginga DF: TV Digital se faz com Ginga! Seção Blog Ginga**, 03 jul. 2009. Disponível em: <<http://www.gingadf.com/blogGinga/?p=129>>.
Acesso em: 05 mai. 2010.

PAES, Alexsandro; ANTONIAZZI, Renato. Padrões de middleware para TV Digital. IN: VII Semana de Engenharia da UFF e IV Seminário Fluminense de Engenharia, 2005 – Niterói. **Anais**.. UFF, 2005. Disponível em: http://www.midiacom.uff.br/itvsoft/pdf/paes_2005.pdf>.
Acesso em: 15 out. 2009. .

PASCHOAL NETO, José Dias; CARVALHO, José Oscar Fontanini. Os novos paradigmas da informação e a tv digital: o papel das TVs universitárias na construção de conteúdos de maneira colaborativa através de redes interdisciplinares . **Informação & Sociedade : Estudos**, João Pessoa, v.18, n.3, p. 59-72, set/dez. 2008. Disponível em:
<<http://www.ies.ufpb.br/ojs2/index.php/ies/article/view/1806/2272>>.
Acesso em: 6 out. 2009.

PASCHOAL NETO, José Dias; CARVALHO, José Oscar Fontanini. Os novos paradigmas da informação e a tv digital: o papel das tvs universitárias na construção de conteúdos de maneira colaborativa através de redes interdisciplinares. **Informação & Sociedade : Estudos**, João Pessoa, v.18, n.3, p. 59-72, set/dez. 2008. Disponível em: <<http://www.ies.ufpb.br/ojs2/index.php/ies/article/view/1806/2272>>. Acesso em: 6 out. 2009.

PINTO, Lady Daiana; QUEIROZ-NETO, José; SILVA, Kátia Cilene. Tecnologias para o desenvolvimento de aplicações educacionais para TV Digital. IN: II Congresso de Pesquisa e Inovação da Rede Nordeste de Educação Tecnológica, 2007, João Pessoa. **Anais...** João Pessoa: CONNEPI, 2007. Disponível em: <http://www.redenet.edu.br/publicacoes/arquivos/20080212_081200_INFO-078.pdf>. Acesso em: 12 out. 2009.

PRESSMAN, Roger S. **Engenharia de Software**. 6. Ed. São Paulo: McGraw-Hill, 2006. 720 p. ISBN 8586804576 (broch.)

QUIZ. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Quiz&oldid=22207630>>. Acesso em: 2 nov. 2010.

REISMAN, Richard R. **Rethinking Interactive TV – I want my Coactive TV**. [S.l.] Teleshuttle Corporation, 2002. Disponível em: <<http://www.teleshuttle.com/cotv/CoTVIntroWtPaper.htm>>. Acesso em 10 abr. 2010.

ROYALTIES. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Royalty&oldid=22187219>>. Acesso em: 2 nov. 2010.

SANTOS, Nelson M. F. et al. Análise dos principais canais de retorno para TV Digital (Resumo). IN: 17º Simpósio Internacional de Iniciação Científica (SIICUSP), 2009, São Paulo. **Anais...** São Paulo: USP, 2009. Disponível em: <http://www.usp.br/siicusp/Resumos/16Siicusp/5419.pdf>>. Acesso em: 02 mar. 2010.

SCHROEDER, Everton. **Desenvolvimento de servidor RSS para TV Digital interativa**. 2007. 73 f. Trabalho de Conclusão de Curso (Bacharelado) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2007. Disponível em: <<http://campeche.inf.furb.br/tccs/2007-II/TCC2007-2-12-VF-EvertonSchroeder.pdf>>. Acesso em: 05 mar. 2010.

SILVA, Jones Quadros. **TV Digital interativa**. 2003. 56 f. Monografia (Especialização) - Curso de Especialização em Redes de Computadores, Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2003. Disponível em: <http://www.wirelessbrasil.org/wirelessbr/colaboradores/jones_quadros/tv_digital.pdf>. Acesso em: 03 mar. 2010.

SILVA SANTOS, Marcos Eduardo et al. Desenvolvendo aplicações Home Care no ambiente Ginga. IN: 35ª InfoBrasil TI & Telecom e II Congresso Tecnológico InfoBrasil, 2009, Fortaleza. **Anais...** Fortaleza: InfoBrasil, 2009. Disponível em: <<http://www.infobrasil.inf.br/iConstructor/Custom/anais2009/Desenvolvendo%20Aplica%C3%A7%C3%B5es%20Home%20Care%20no%20Ambiente%20GINGA.pdf>>. Acesso em: 12 out. 2009.

SMART CARD. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Smart_card&oldid=21927897>. Acesso em: 2 nov. 2010.

SOARES, Luiz Fernando Gomes. As múltiplas possibilidades do middleware Ginga. **Produção Profissional: Revista de comunicação e técnica audiovisual**, São Paulo, p. 76-83, jun. 2008. Disponível em: <ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/journalpapers/2008_06_soares.pdf>. Acesso em: 7 out. 2009.

_____. Middleware Ginga. **Produção Profissional: Revista de comunicação e técnica audiovisual**, São Paulo, p. 68-75, jun. 2008. Disponível em: <<http://www.gingancl.org.br/resources/SBTVD-Ginga.pdf>>. Acesso em: 4 out. 2009.

SOM SURROUND. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Som_surround&oldid=22275099>. Acesso em: 2 nov. 2010.

SOUZA, Raquel Araújo; BORGES, Fabio. **O padrão de criptografia simétrica AES**. 2006. Artigo apresentado no Programa de Iniciação Científica do Centro de Educação Superior à Distância do Estado do Rio de Janeiro - Licenciatura em Matemática. Rio de Janeiro, 2006. Disponível em: <<http://www.lncc.br/~borges/doc/O%20padr%C3%A3o%20de%20criptografia%20sim%C99trica%20AES.pdf>>. Acesso em: 10 ago. 2010

TABORDA, Caio et al. **TV Digital no Brasil: impactos para as emissoras e consumidores**. Apresentação final – Programa de imersão rápida em Consultoria. São Paulo, AMCHAM Brasil, 2007. Disponível em: < <http://tvdigitalnobrasil.files.wordpress.com/2007/10/projeto-tv-digital-amcham-bah.pdf>>. Acesso em 15 out. 2009.

TAVARES, Tatiana Aires et al. A TV Digital interativa como ferramenta de apoio à educação infantil. **Revista Brasileira de Informática na Educação**, Porto Alegre, Vol. 15, n. 2, mai./ago. 2007. Disponível em: < <http://bibliotecadigital.sbc.org.br/download.php?paper=939>>. Acesso em: 10 out. 2009.

TIC 2008. Proporção de domicílios com acesso à Internet. **Centro de Estudos sobre as Tecnologias da Informação e da Comunicação - CETICbr**. Seção Pesquisas e Indicadores, set./nov. 2008. Disponível em: < <http://www.cetic.br/usuarios/tic/2008-total-brasil/rel-geral-04.htm>>. Acesso em: 22 mai. 2010.

WOOFER. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2010. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Woofer&oldid=21295024>>. Acesso em: 2 nov. 2010.

ZANCANARO, Airton; SANTOS, Paloma Maria; TODESCO, Jose Leomar. Ginga-J ou Ginga-NCL: características das linguagens de desenvolvimento de recursos interativos para a TV Digital. IN: 1º Simpósio Internacional de Televisão Digital (SIMTVD), 2009, Bauru. **Anais...** Bauru: UNESP, 2009. Disponível em: <<http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/ZANCANARO;%20SANTOS;%20TODESCO%20-%20GINGA-J%20OU%20GINGA-NCL%20%281084-1108%29.pdf>>. Acesso em: 15 mar. 2010.

APÊNDICE A – CASOS DE USO NO MODO EXPANDIDO

Nome do Caso de Uso	C01: Navegar pela aplicação
Caso de uso geral	
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a navegação entre os itens da tela, onde o usuário utiliza os botões do controle remoto e o sistema responde de acordo com a função selecionada.
Pré-condições	
Pós-condições	
Ações do ator	Resposta do sistema
1 – Pressiona a tecla desejada do controle remoto virtual	2 – Verifica a tecla pressionada
	3 – Responde com o evento correspondente
	4 – Encerra operação
Restrições / Validações	

Nome do Caso de Uso	C02: Listar itens de uma categoria
Caso de uso geral	Navegar pela aplicação
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação de listagem de itens de uma categoria de produtos selecionada pelo usuário para exibição.
Pré-condições	
Pós-condições	
Ações do ator	Resposta do sistema
1 – Seleciona a categoria desejada e pressiona o botão de ação	
	2 – Requisita junto ao servidor a lista de produtos da categoria escolhida.
	3 – Exibe a lista no painel de visualização dos itens.
	4 – Passa o foco de entrada para o primeiro item do painel de itens.
	5 – Encerra operação
Restrições / Validações	

Nome do Caso de Uso	C03: Exibir detalhes de um item específico
Caso de uso geral	Listar itens da categoria
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação de exibição de detalhes de um item selecionado no painel de itens da categoria.
Pré-condições	Ao menos uma categoria deve ter sido selecionada anteriormente para listagem de itens no painel.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Seleciona o item desejado e pressiona o botão de ação	
	2 – Requisita junto ao servidor os dados referentes ao produto selecionado.
	3 – Exibe os dados retornados no painel de exibição do item.
	4 – Passa o foco de entrada para o painel do item.
	5– Encerra operação
Restrições / Validações	

Nome do Caso de Uso	C04: Adicionar itens ao carrinho de compras
Caso de uso geral	Efetuar Login
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário adiciona um item ao carrinho de compras.
Pré-condições	Um item deve estar sendo exibido no painel de detalhes de item.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Ajusta a quantidade desejada do item;	
2 – Seleciona a opção Adicionar Item ao carrinho;	
	3 – Se o cliente não estiver logado, executar o caso de uso C05: efetuar login/autenticação;
	4 – Caso o item já esteja no carrinho, exclui o anterior;
	5 – Adiciona o item ao carrinho;
	6– Atualiza o status do carrinho na tela;
	7 – Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C05: Efetuar login/autenticação
Caso de uso geral	Navegar pela aplicação
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário se autentica para poder adicionar itens no carrinho de compras.
Pré-condições	O usuário deve estar cadastrado no servidor remoto.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Informa login e senha;	
	2 – Valida login e senha, em consulta ao servidor remoto;
	3 – Se o login for válido e uma operação de adição ao carrinho estiver em andamento, executa o caso de uso C04:Adicionar itens ao carrinho;
	4 – Se o login for inválido, solicita ao usuário que acesse a página de cadastro (à parte), caso de uso C11:cadastro de usuário na página web;
	5 – Se o login for válido, atualiza o status do usuário na tela ;
	6– Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C06: Consultar carrinho de compras
Caso de uso geral	
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário consulta os itens adicionados ao carrinho de compras.
Pré-condições	1 – O usuário deverá estar logado; 2 - Ao menos um item deve ter sido adicionado ao carrinho.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário seleciona a opção Visualizar Carrinho	
	2 – Pesquisar os itens adicionados ao carrinho;
	3 – Exibe em uma janela os itens existentes no carrinho, quantidade reservada e valor total;
	4 – Permite ao cliente excluir itens ou modificar a quantidade reservada, executando o caso de uso C07:excluir/modificar item do carrinho de compras;
	5– Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C07: Excluir/modificar item do carrinho de compras
Caso de uso geral	C06: Consultar carrinho de compras
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário modifica a quantidade reservada ou exclui um item do carrinho de compras.
Pré-condições	1 – O usuário deverá estar logado; 2 - Ao menos um item deve ter sido adicionado ao carrinho; 3 – O item em questão deve estar selecionado na tela de visualização do carrinho de compras.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário seleciona a opção de alterar quantidade ou excluir o item;	
	2 – O sistema ajusta a quantidade ou exclui o item do carrinho;
	3 – O sistema atualiza o status do carrinho no painel correspondente da tela;
	4 – Permite ao cliente excluir itens ou modificar a quantidade reservada;
	5– Encerra operação.
Restrições / Validações	1 – A quantidade não poderá ser menor do que 1;

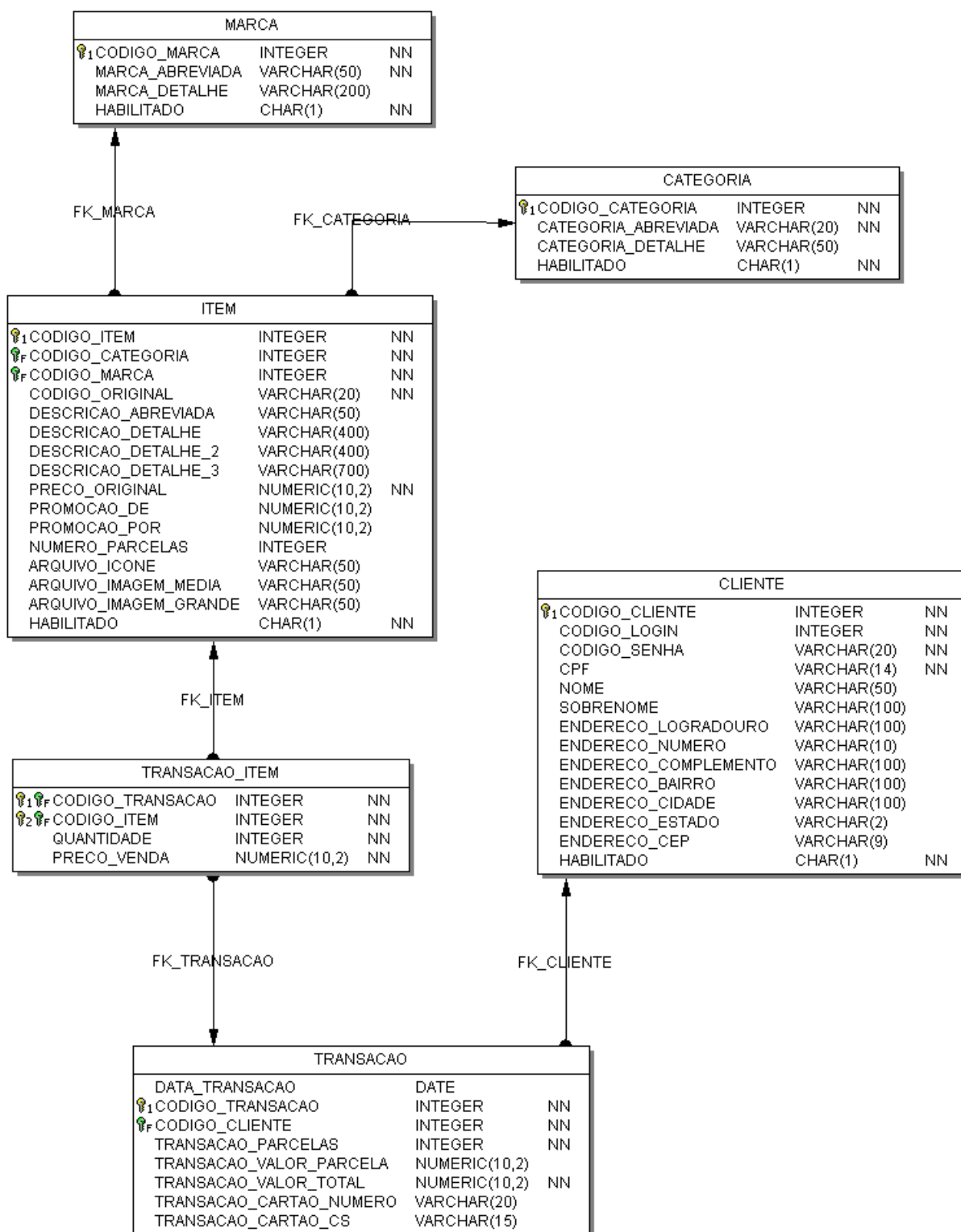
Nome do Caso de Uso	C08: Cancelar carrinho de compras
Caso de uso geral	Adicionar itens no carrinho de compras
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário exclui o carrinho de compras de uma só vez, por completo.
Pré-condições	1 – O usuário deverá estar logado; 2 - Ao menos um item deve ter sido adicionado ao carrinho.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário seleciona a opção de excluir carrinho de compras;	
	2 – O sistema elimina os itens adicionados ao carrinho e quaisquer informações relacionadas;
	3 – O sistema atualiza o status/informações do carrinho no painel correspondente da tela;
	4– Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C09: Efetuar a compra dos itens do carrinho
Caso de uso geral	
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário efetua a operação de compra dos itens adicionados ao carrinho, concluindo a operação de pedido.
Pré-condições	1 – O usuário deverá estar logado; 2 - Ao menos um item deve ter sido adicionado ao carrinho.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário seleciona a opção de comprar itens do carrinho/concluir pedido;	
	2 – Exibe um resumo dos itens do carrinho, bem como dados de identificação do cliente;
	3 – Exibe as opções de forma de pagamento;
4 – Cliente preenche dados requisitados e forma de pagamento;	
	5 – Sistema pede confirmação ou cancelamento;
6 – Usuário escolhe a opção;	
	7– Caso cancelar, volta à tela principal;
	8 – Caso confirmar, avisa o usuário, zera o carrinho e atualiza informações na tela;
	9– Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C10: Efetuar logout
Caso de uso geral	
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário efetua o logout do sistema (desconecta-se).
Pré-condições	O usuário deverá estar logado.
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário seleciona a opção de logout;	
	2 – Caso haja itens no carrinho de compras, salva os dados no servidor, vinculando ao usuário conectado;
	3 – Elimina quaisquer dados de sessão do usuário armazenados localmente;
	4 – Atualiza informações na tela;
	5 – Encerra operação.
Restrições / Validações	

Nome do Caso de Uso	C11: Cadastro de usuário
Caso de uso geral	
Ator principal	Usuário
Atores secundários	
Resumo	Descreve a operação onde o usuário cadastra-se na página específica, para poder autenticar-se e efetuar compras.
Pré-condições	
Pós-condições	
Ações do ator	Resposta do sistema
1 – Usuário digita o endereço da página no navegador Web;	
	2 – Exibe boas-vindas e formulário para preenchimento de dados do usuário;
	3 – Verifica se o CPF digitado no formulário já existe no sistema;
	4 – Caso exista, avisa o usuário, mas não permite alteração;
	5 – Não existindo, salva o registro;
	6 – Retorna um código de usuário numérico para ser utilizado no login da aplicação de compras;
	7– Encerra operação.
Restrições / Validações	Esta operação será executada à parte em relação à aplicação Xlet; o usuário irá acessar a página de cadastro via navegador Web e efetuar o cadastro.

APÊNDICE B – DIAGRAMA E-R DO BANCO DE DADOS DO SERVIDOR



APÊNDICE C – INSTALANDO E CONFIGURANDO O AMBIENTE

Neste apêndice são descritos os passos para a configuração do ambiente necessário à execução e testes com o protótipo desenvolvido. Recomenda-se que os passos sejam executados na ordem apresentada a seguir.

INSTALANDO O SGBD FIREBIRD

1. A versão do sistema gerenciador de banco de dados (SGBD) *Firebird* utilizada foi a 2.0.4 (Win32 *Build*). Neste caso, para evitar quaisquer conflitos de versão relacionados às bibliotecas do *driver* de acesso JDBC utilizadas no projeto (embutidas no projeto do servidor *web*), recomenda-se a instalação a partir da versão disponibilizada no CD do projeto;
2. Vá até o diretório *instalacao* do CD do projeto;
3. Clique duas vezes no arquivo *Firebird-2.5.0.26074_1_Win32.exe* para executar o programa de instalação do SGBD;
4. Siga os passos solicitados, até o final da instalação;
5. Após encerrar, acesse a interface de controle do servidor (*Painel de Controle* do Windows / *Firebird 2.0 Server Manager*);
6. Caso o serviço esteja parado, clique em Start, para iniciar a execução do SGBD.

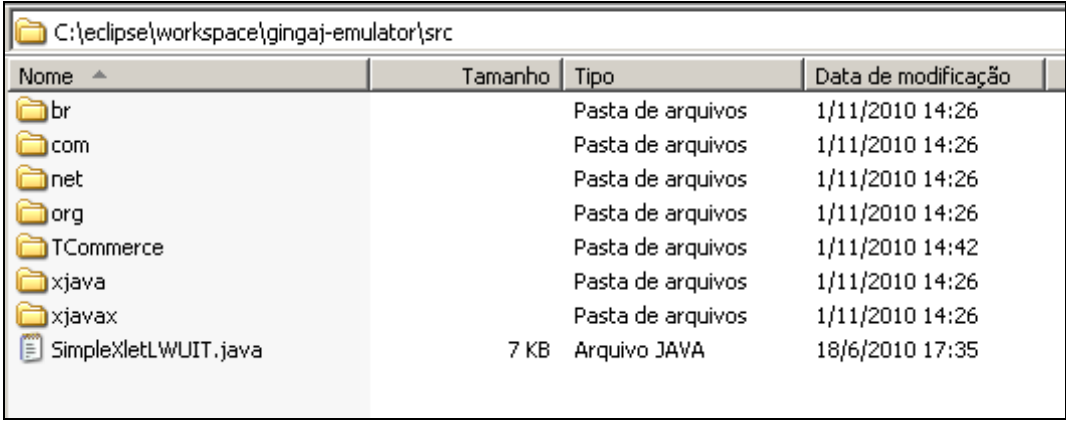
CONFIGURANDO O SERVIDOR WEB NA IDE *NetBeans*

1. A versão da IDE *NetBeans* utilizada foi a 6.9.1, na versão Java completa, que dá suporte ao desenvolvimento de aplicações da plataforma web , Java SE, Java ME, Java *Enterprise* e Java *Card Connected*, além de trazer embutidos os servidores de aplicação *Glassfish Server Open Source Edition 3.0.1* e *Apache Tomcat 6.0.26*. A última versão da IDE pode ser baixada no site oficial: <http://netbeans.org/downloads/index.html> . Caso preferir, instale a partir da cópia disponibilizada na pasta *instalacao* do CD do projeto;
2. Após instalar o IDE, vá até o diretório *fontes* do CD do projeto e copie a pasta *TCommerceServer*;
3. Cole a pasta selecionada **na raiz (c:)** do sistema operacional ;
4. Abra a IDE *NetBeans* e acesse o menu *Arquivo / Abrir Projeto..*;
5. Localize a pasta do projeto copiada em C: (passo 3) , selecione com o mouse e clique em *Abrir Projeto*;
6. Aguarde a IDE completar a tarefa de abrir o projeto;
7. Clique com o botão direito na raiz do projeto e selecione a opção *Executar*;
8. Irá abrir uma página do navegador contendo o texto “Servidor web TCommerce Server em execução e aguardando requisições..”.

CONFIGURANDO O GINGA-J EMULADOR NA IDE Eclipse

1. A importação do projeto do emulador para dentro do Eclipse é feita com auxílio do *plugin EGIT*, disponível para as versões 3.4 ou mais recentes da IDE;
2. Caso já tenha uma versão da IDE 3.4 ou mais recente previamente instalada e queira instalar o plugin EGIT nela, continue a partir do próximo passo (passo 3). Caso queira utilizar a versão Eclipse 3.5.0 Galileo disponibilizada no CD, copie a pasta *eclipse* existente no diretório *instalacao* do CD e cole na raiz do sistema operacional. Esta pasta contém uma cópia da IDE já instalada e com o plugin EGIT já configurado, bastando clicar duas vezes no arquivo *eclipse.exe* para iniciar a execução do ambiente IDE. Após, vá para o passo 9 e siga o restante;
3. Com o Eclipse aberto, vá ao menu *Help / Install New Software..*;
4. Clique em *Add..*
5. Preencha *Name* com *JG it* e *Location* com *http://download.eclipse.org/egit/updates* ;
6. Marque a opção *Eclipse Git Feature* e clique em *Next*;
7. Clique em *Finish*;
8. Aguarde pela finalização da instalação e reinicie o Eclipse;
9. Agora é necessário importar o projeto do emulador para dentro do Eclipse. Embora seja possível baixar a versão mais atual do repositório Ginga CDN utilizando a ferramenta GIT, este procedimento não será feito, em virtude de o protótipo ter sido desenvolvido com base na versão do emulador disponibilizada em 18/06/2010 pelo repositório. Esta versão está disponibilizada na pasta *instalacao* do projeto;
10. Copie a pasta *gingaj-emulador* que está no diretório *instalacao* do CD e cole no diretório *workspace* do Eclipse;
11. Com o Eclipse aberto, vá ao menu *File / Import..*;

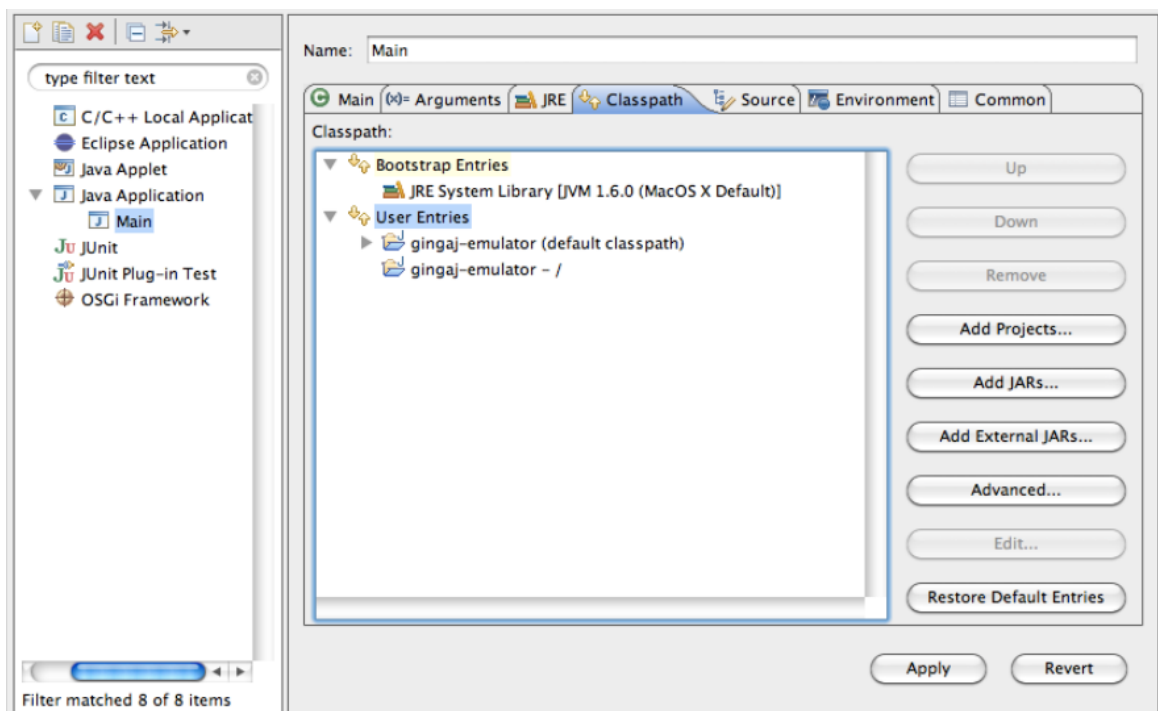
12. Selecione a opção *Git / Projects from Git* e clique em *Next*;
13. Na tela *Select a Git Repository*, selecione a opção *gingaj-emulador* que aparece na lista. Caso não apareça, clique no botão *Add* e localize o diretório onde a pasta foi colada (passo 10);
14. Após selecionar na lista do passo anterior, clique em *Next*;
15. Na tela *Select a Wizard and decide how to share the imported projects* deixe marcadas as opções *Import Existing Projects* e *Try to share newly created projects automatically* e clique em *Next*;
16. Na tela *Import Projects* marque a opção existente na lista e clique em *Finish*;
17. Pelo Windows Explorer, copie a pasta *TCommerce* presente no diretório *fontes* do CD e cole dentro da pasta *src* do projeto *gingaj-emulador*, que se encontra no diretório *workspace* do Eclipse. A lista de pastas deve ficar semelhante à figura abaixo:



Nome	Tamanho	Tipo	Data de modificação
br		Pasta de arquivos	1/11/2010 14:26
com		Pasta de arquivos	1/11/2010 14:26
net		Pasta de arquivos	1/11/2010 14:26
org		Pasta de arquivos	1/11/2010 14:26
TCommerce		Pasta de arquivos	1/11/2010 14:42
xjava		Pasta de arquivos	1/11/2010 14:26
xjavax		Pasta de arquivos	1/11/2010 14:26
SimpleXletLWUIT.java	7 KB	Arquivo JAVA	18/6/2010 17:35

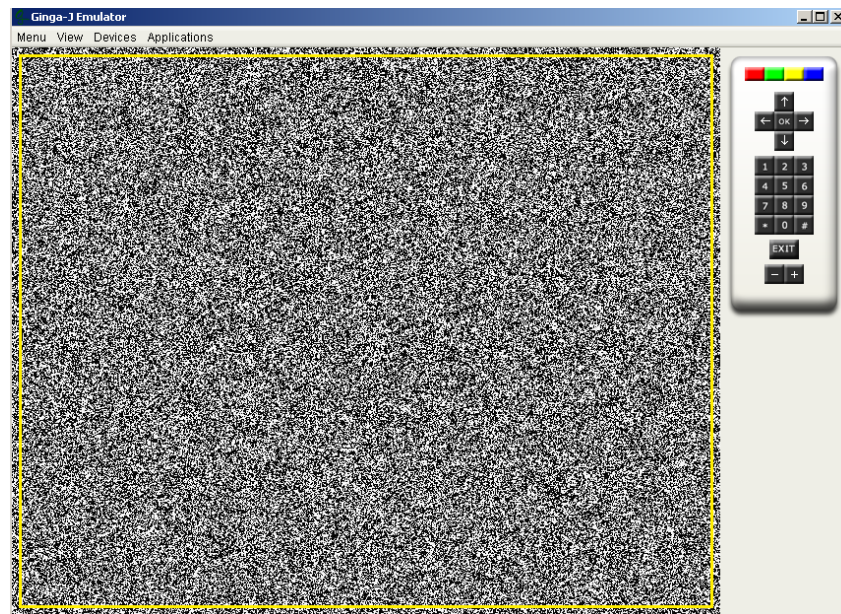
18. Dentro do Eclipse, clique com o botão direito na raiz do projeto *gingaj-emulador* e selecione a opção *Refresh*;
19. Clique com o botão direito na raiz do projeto e selecione a opção *Run As / Java Application...*;

20. Selecione a classe *Main* – *net.beiker.xletview* (que é a classe principal do emulador, a partir do qual será executado) e clique no botão OK;
21. O projeto será compilado e executado com uma mensagem de erro;
22. Então, com o botão direito na raiz do projeto e selecione a opção *Run As / Run Configurations..*;
23. Na janela aberta escolha no menu lateral esquerdo o item "Java Application ->Main " e selecione a aba *Classpath* .;
24. Depois clique em *User Entries* e clique no botão *Advanced..*;
25. Selecione *Add Folders* e no pop-up aberto escolha o diretório com o nome do projeto do emulador importado para o Eclipse – opção *gingaj-emulator* que irá aparecer na lista Clique em OK;
26. O resultado deverá ser semelhante ao da figura abaixo:



27. Clique em *Apply* e depois em *Run*;

28. O emulador irá executar com sucesso e apresentar a tela abaixo:



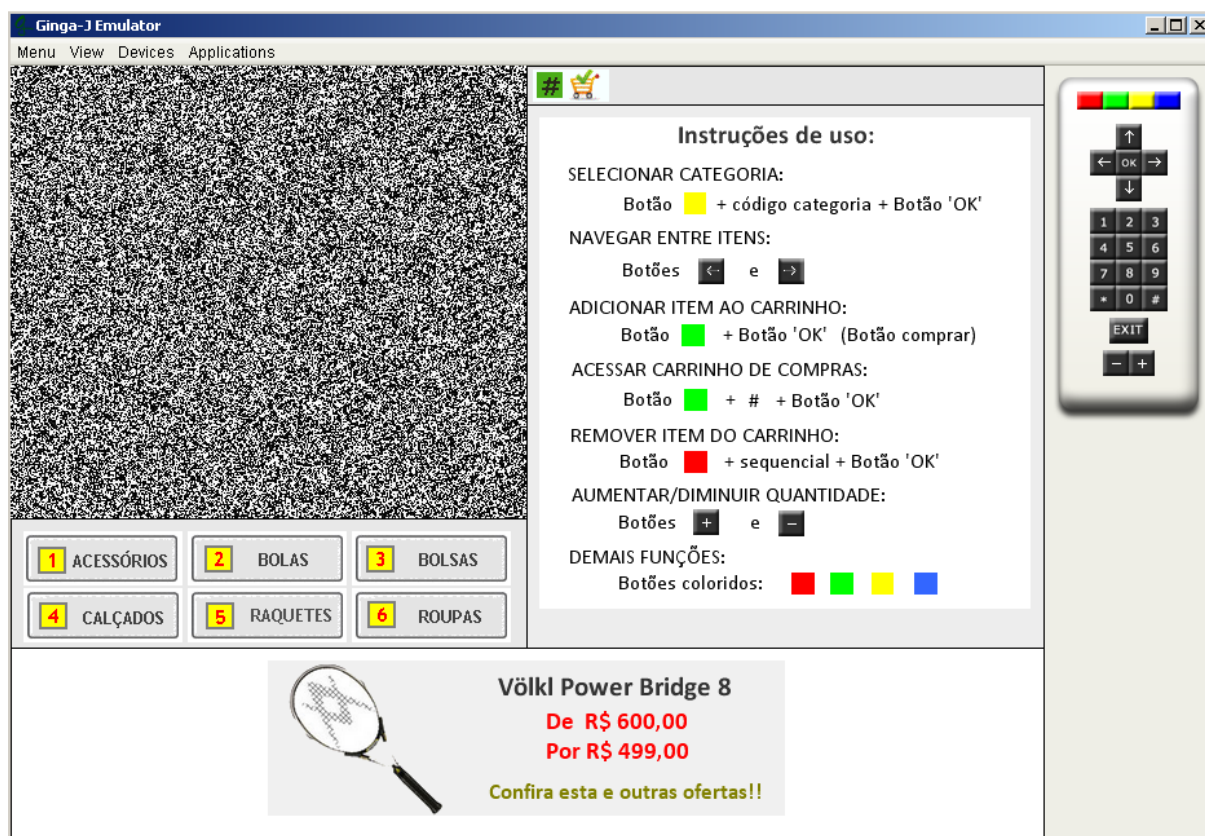
EXECUTANDO O PROTÓTIPO NO EMULADOR

1. Vá ao diretório *TCommerce* copiado para dentro da pasta *src* do emulador no passo 17 do item anterior – deve estar em `../Eclipse/workspace/gingaj-emulator/src/TCommerce` - e edite o arquivo `xmlproperties.xml`, informando o endereço ip ou nome de rede do servidor web *TCommeceServer* e a porta ouvinte. Caso o servidor seja executado na máquina local, configure o parâmetro *Server* com o valor `localhost`;
2. Com o emulador executando, selecione o menu *Applications > Manage Applications*, na tela do emulador;
3. Clique na pasta *Default group* e em seguida selecione o botão *New application*;
4. Selecione a aplicação criada na árvore do grupo digite um nome para ela, no campo *Name*;

5. No campo *path*, clique no botão e selecione o diretório *bin* do projeto do emulador – normalmente no diretório *workspace* do Eclipse: `.../Eclipse/workspace/gingaj-emulator/bin`;
6. Agora será necessário indicar o nome da classe principal do xlet. No campo Xlet **digite exatamente**: `TCommerce.gui.XletPrincipal`
7. Clique em *Save & Close*;
8. Agora vá ao menu Applications e selecione a aplicação com o nome definido no passo 3;
9. O Xlet irá executar e apresentar a tela inicial com as instruções de uso e ofertas.

APÊNDICE D – INSTRUÇÕES DE USO DO PROTÓTIPO

Ao iniciar a execução, o protótipo exibirá um painel com algumas instruções de uso, conforme se pode observar na figura a seguir:



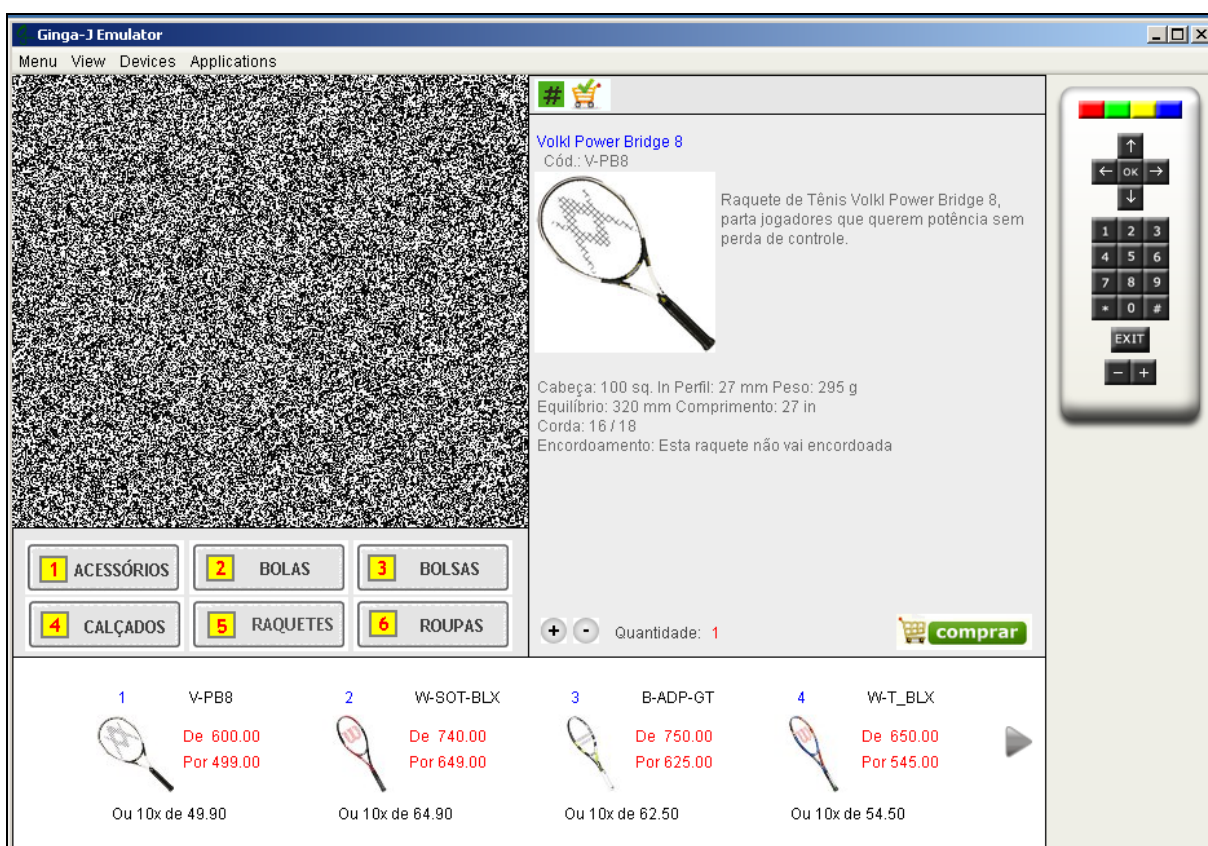
Na sequência, serão listados os casos de uso implementados correspondentes às operações do protótipo e as instruções para utilização do controle na tarefa de interação entre o usuário e o Xlet em cada operação.

LISTAR ITENS DE UMA CATEGORIA

Como se pode observar na figura anterior, os itens de compra foram agrupados em

seis categorias distintas, cada qual numerada no intervalo de 1 a 6. A primeira operação do usuário no protótipo será a seleção de uma categoria, para que os itens cadastrados nesta sejam exibidos no painel inferior de forma resumida, podendo ser percorridos e exibidos individualmente de forma mais detalhada. Logo, a sequência utilizada para seleção de itens é:

- a) pressione a tecla amarela do controle remoto;
- b) em seguida, pressione o número correspondente à categoria desejada;
- c) pressione OK no controle remoto e os itens serão exibidos como abaixo.



EXIBIR DETALHES DE UM ITEM ESPECÍFICO

Logo uma categoria é selecionada, os detalhes do primeiro item desta são exibidos no painel superior direito. Para navegar entre os itens, a sequência utilizada é:

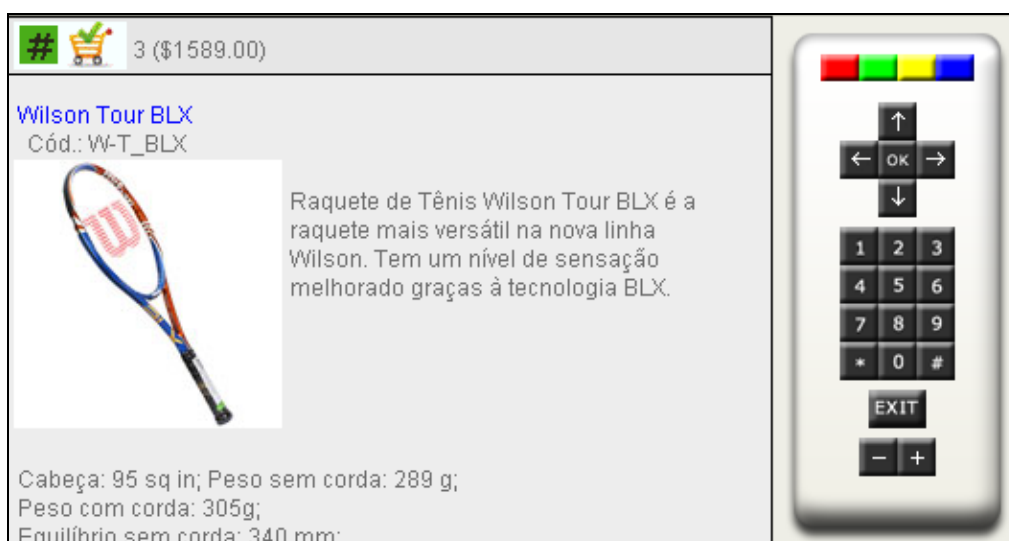
- a) caso no painel horizontal inferior haja um ícone de seta para a esquerda, há mais itens anteriores a este. Pressione a tecla ← do controle remoto;
- b) caso haja um ícone de seta para a direita, há mais itens adiante para serem exibidos. Pressione a tecla → do controle remoto.

ADICIONAR ITENS AO CARRINHO DE COMPRAS

Assim que um item é exibido, pode ser adicionado ao carrinho de compras. Caso um item 'X' já conste no carrinho, uma nova adição deste será permitida. Entretanto, a operação anterior será sobreposta pela mais atual, como se nunca tivesse existido. A sequência para adicionar ao carrinho é a seguinte:

- a) ajuste a quantidade desejada pressionando as teclas + ou - do controle remoto. Por padrão, a quantidade inicial é definida como 1;
- b) pressione a tecla verde do controle remoto;
- c) pressione a tecla OK do controle remoto, e o item será adicionado ao carrinho.

Ao lado do ícone de carrinho irão aparecer as informações de quantidade de itens no carrinho e total em dinheiro correspondente:



CONSULTAR/ACESSAR O CARRINHO DE COMPRAS

O carrinho de compras pode ser acessado, e seus itens poderão ser visualizados e até mesmo excluídos, um a um ou todos de uma vez. Por outro lado, a partir do carrinho de compra é que as operações de *login* e finalização da compra serão iniciadas. A sequência para acesso ao carrinho é listada abaixo:

- a) em qualquer tela, pressione a tecla verde do controle remoto;
- b) pressione a tecla # do controle remoto;
- c) pressione a tecla OK do controle remoto, e os itens do carrinho serão listados:



The screenshot shows a shopping cart interface on a screen. At the top left, there is a green '#' icon and a shopping cart icon, followed by the text '11 (\$4515.00)'. Below this, the text 'ITENS SELECIONADOS:' is displayed. A table lists the selected items with columns for 'PRODUTO', 'QUANT.', and 'TOTAL'. The items are numbered 1 through 6. At the bottom of the table, the total is shown as 'TOTALS: 11 R\$ 4.515,00'. Below the table, there are three buttons: 'comprar mais' (blue), 'zera carrinho' (red), and 'finalizar' (green). To the right of the screen, a remote control is overlaid, showing a directional pad, an 'OK' button, a numeric keypad, an 'EXIT' button, and '-' and '+' buttons.

	PRODUTO	QUANT.	TOTAL
1	Volk Power Bridge 8	1	R\$ 499,00
2	Wilson Tour BLX	2	R\$ 1.090,00
3	Puma Voltaic II Cinza	1	R\$ 190,00
4	Tênis Adidas Barricade VWomen	2	R\$ 890,00
5	Prokennex Destiny Ti Black	4	R\$ 1.196,00
6	Yonex RDIS100 MP	1	R\$ 650,00
TOTALS:		11	R\$ 4.515,00

EXCLUIR ITEM DO CARRINHO DE COMPRAS

Exclui-se itens de forma individual do carrinho de compras conforme a sequência:

- na tela do carrinho, pressione a tecla amarela do controle remoto;
- pressione o número de sequência do item na lista (a exemplo da figura acima);
- pressione a tecla OK do controle remoto. O item da sequência correspondente será removido e as informações do carrinho serão atualizadas.

ZERAR CARRINHO DE COMPRAS

A sequência abaixo permite eliminar todos os itens do carrinho de uma só vez:

- na tela do carrinho, pressione a tecla vermelha do controle remoto;

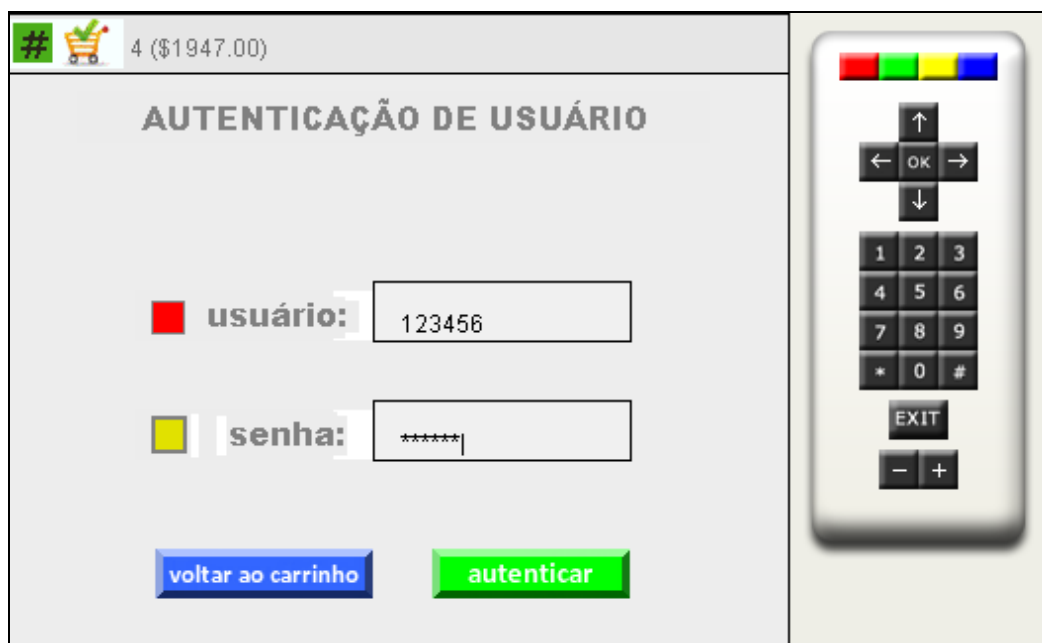
Os itens serão todos removidos com esta operação única, sem exigir o pressionamento da tecla OK. O carrinho será então atualizado:



EFETUAR LOGIN/AUTENTICAÇÃO

A partir do carrinho de compras, corresponde à operação do “botão” *Finalizar*:

- a) na tela do carrinho, pressione o botão verde do controle remoto. Será exibida a tela de autenticação do usuário:



- b) para testar o protótipo utilize: usuário=123456; senha=654321;
- c) para digitar o código de usuário, pressione o botão vermelho do controle;
- d) pressione as teclas numéricas para informar o código de usuário;
- e) pressione o botão amarelo para passar ao campo de senha;
- f) pressione as teclas numéricas correspondentes à senha;
- g) pressione o botão verde do controle remoto para autenticar no servidor web.

Caso o usuário esteja cadastrado, será exibida a tela final da operação, com seus dados pessoais, possibilidade de parcelamento da compra e campos de entrada para digitação do número e código de segurança do cartão de crédito do cliente:

4 (\$1947,00)

DADOS PESSOAIS :
 NOME : Franco Bortolotto
 CPF : 025.708.139-90

ENDEREÇO :
 Rua 14 de Julho, s/n
 São Bento Baixo - 88867-000
 Nova Veneza - SC

SUA COMPRA :
 ITENS SELECIONADOS: R\$ 1.947,00
 DESPESAS DE FRETE: R\$ 10,00
TOTAL DA TRANSAÇÃO: R\$ 1.957,00

+ - 10 X de R\$ 195,70

■ Número do cartão de crédito: 4521369688963541

■ Código de segurança : 518|

[voltar ao carrinho](#) [finalizar](#)

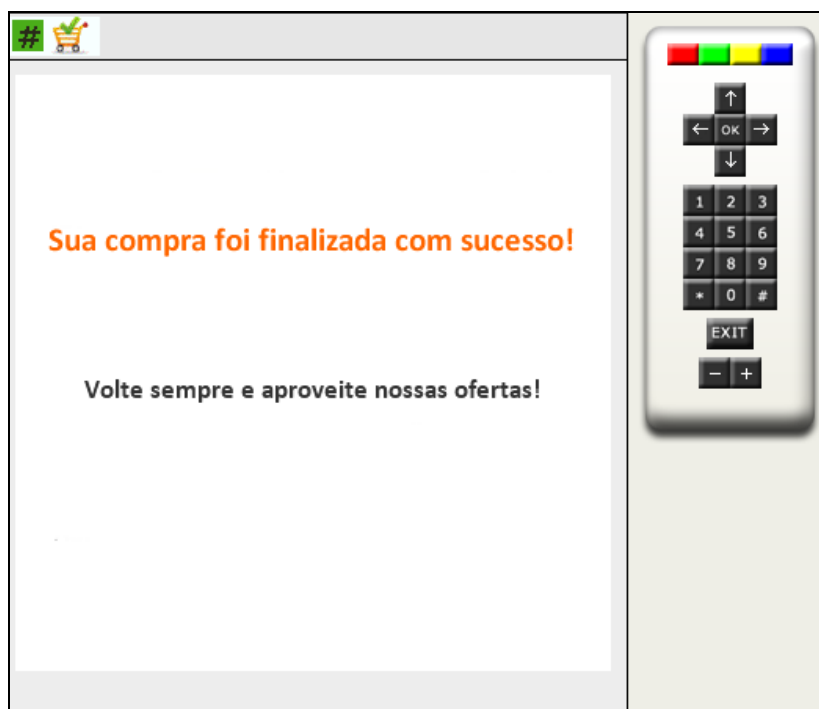
The keypad on the right includes: a red key, a green key, a yellow key, a blue key, an OK key, directional arrows, a numeric keypad (1-9, *, 0, #), an EXIT key, and +/- keys.

EFETUAR A COMPRA/ENCERRAR TRANSAÇÃO

A sequência para finalização da transação é a seguinte:

- confira se os dados estão corretos;
- escolha o número de parcelas, utilizando os botões + e – do controle;
- pressione a tecla vermelha do controle para entrar com o número de cartão;
- pressione as teclas numéricas correspondentes ao número do cartão (16 posições numéricas);
- pressione a tecla amarela do controle para passar à digitação do campo de código de segurança;
- pressione as teclas numéricas correspondentes;
- pressione a tecla verde do controle remoto, correspondente à operação *Finalizar*.

Os dados da transação são enviados ao servidor web para serem inseridos na base de dados. Caso a operação seja concluída com sucesso, a seguinte tela será exibida:



Nesse instante, o caso de uso C10 – *efetuar logout* será executado automaticamente de maneira implícita, sem que o usuário seja solicitado para realizar tal operação. Após 5 segundos, a aplicação assumirá o estado inicial, reiniciando o ciclo com a exibição do painel de instruções e ofertas em anúncio.

ANEXO A – LISTA DE PACOTES MÍNIMOS DO GINGA-J (ABNT-NBR 15606-4)

Pacotes da plataforma Java:

- java.awt
- java.awt.color
- java.awt.event
- java.awt.font
- java.awt.im
- java.awt.image
- java.beans
- java.io
- java.lang
- java.lang.ref
- java.lang.reflect
- java.math
- java.net
- java.rmi
- java.rmi.registry
- java.security
- java.security.acl
- java.security.cert
- java.security.interfaces
- java.security.spec
- java.text
- java.util
- java.util.jar
- java.util.zip
- javax.microedition.io
- javax.microedition.pki
- javax.microedition.xlet
- javax.microedition.xlet.ixc
- javax.security.auth.x500

Pacotes da especificação JavaTV 1.1 e JMF 1.0:

- javax.media
- javax.media.protocol
- javax.tv.graphics
- javax.tv.locator
- javax.tv.media
- javax.tv.net
- javax.tv.service
- javax.tv.service.guide
- javax.tv.service.navigation
- javax.tv.service.selection
- javax.tv.service.transport
- javax.tv.util
- javax.tv.xlet

Pacotes da especificação JAVADTV 1.3:

- com.sun.dtv.application
- com.sun.dtv.broadcast
- com.sun.dtv.broadcast.event
- com.sun.dtv.filtering
- com.sun.dtv.io
- com.sun.dtv.locator
- com.sun.dtv.lwuit
- com.sun.dtv.lwuit.animations
- com.sun.dtv.lwuit.events
- com.sun.dtv.lwuit.geom
- com.sun.dtv.lwuit.layouts
- com.sun.dtv.lwuit.list
- com.sun.dtv.lwuit.painter
- com.sun.dtv.lwuit.plaf
- com.sun.dtv.lwuit.util
- com.sun.dtv.media
- com.sun.dtv.media.audio
- com.sun.dtv.media.control
- com.sun.dtv.media.dripfeed
- com.sun.dtv.media.format
- com.sun.dtv.media.language
- com.sun.dtv.media.text
- com.sun.dtv.media.timeline
- com.sun.dtv.net
- com.sun.dtv.platform
- com.sun.dtv.resources
- com.sun.dtv.security
- com.sun.dtv.service
- com.sun.dtv.smartcard
- com.sun.dtv.test
- com.sun.dtv.transport
- com.sun.dtv.tuner
- com.sun.dtv.ui
- com.sun.dtv.ui.event

Pacotes da especificação JSSE 1.0.1:

- com.sun.net.ssl
- javax.net
- javax.net.ssl
- javax.security.cert

Pacotes da especificação JCE 1.0.1:

- javax.crypto
- javax.crypto.interfaces
- javax.crypto.spec

Pacotes da especificação SATSA 1.0.1:

- javax.microedition.apdu

Pacotes específicos Ginga-J:






- br.org.sbtvd.net
- br.org.sbtvd.net.si
- br.org.sbtvd.net.tuning
- br.org.sbtvd.bridge
- br.org.sbtvd.ui

ANEXO B - PACOTES DTV UI (LWUIT) – STATUS DE IMPLEMENTAÇÃO

Intro » Status atual »

Pacotes DTV UI (lwuit)

Legenda: Status de Implementação

Status	
Não iniciada	
Incompatível com a Norma	
Em construção	
Implementada com Bugs	
Implementada	

Listagem dos Pacotes









com.sun.dtv.lwuit

Status: 

Interfaces

Painter

Classes

Nome da Classe	Status	Comentários
AWTComponent.java		
Button.java		
ButtonGroup.java		
ButtonGroup.java		
Calendar.java		
CheckBox.java		
ComboBox.java		
Command.java		
Component.java		
Container.java		
Dialog.java		
Font.java		
Form.java		
Graphics.java		
Image.java		
Label.java		
List.java		
MediaComponent.java		
RadioButton.java		
StaticAnimation.java		
TabbedPane.java		
TextArea.java		
TextField.java		

Exceções

Não se aplica




com.sun.dtv.lwuit.animations

Status: 

Interfaces

Animation

Classes

Nome da Classe	Status	Comentários
CommonTransitions.java		
Motion.java		
Transition.java		

Exceções

Não se aplica


com.sun.dtv.lwuit.events

Status: 

Interfaces

ActionListener, DataChangeListener, FocusListener, SelectionListener, StyleListener

Classes

Nome da Classe	Status	Comentários
ActionEvent.java		

Exceções

Não se aplica




com.sun.dtv.lwuit.geom

Status: 

Interfaces

Não se aplica

Classes

Nome da Classe	Status	Comentários
Dimension.java		
Point.java		
Rectangle.java		

Exceções

Não se aplica









com.sun.dtv.lwuit.layouts

Status: 

Interfaces

Não se aplica

Classes

Nome da Classe	Status	Comentários
BorderLayout.java		
BoxLayout.java		
CoordinateLayout.java		
FlowLayout.java		
GridLayout.java		
GroupLayout.java		
Layout.java		
LayoutStyle.java		

Exceções

Não se aplica



com.sun.dtv.lwuit.list

Status: 

Interfaces

ListCellRenderer, ListModel

Classes

Nome da Classe	Status	Comentários
DefaultListCellRenderer.java		
DefaultListModel.java		



Exceções

Não se aplica

com.sun.dtv.lwuit.painterStatus: **Interfaces**

Não se aplica

Classes

Nome da Classe	Status	Comentários
BackgroundPainter.java		
PainterChain.java		






Exceções

Não se aplica

com.sun.dtv.lwuit.plafStatus: **Interfaces**

Não se aplica

Classes

Nome da Classe	Status	Comentários
Border.java		
DefaultLookAndFeel.java		
LookAndFeel.java		
Style.java		
UIManager.java		



Exceções

Não se aplica

com.sun.dtv.lwuit.utilStatus: **Interfaces**

Não se aplica

Classes

Nome da Classe	Status	Comentários
Log.java		
Resources.java		

Exceções

Não se aplica

APÊNDICE E – ARTIGO

Protótipo de Aplicação *TCommerce* baseado no Subsistema Procedural do Middleware Ginga – Ginga-J

Franco Bortolotto¹, Paulo João Martins², Fabrício Giordani²

¹Acadêmico do curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC - Brasil

²Professor do curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

francobortolotto@yahoo.com.br, pjm@unesc.net, fgiordani@gmail.com

Resumo: *A Televisão Digital aberta está em foco nos países emergentes. Além dos benefícios relacionados à imagem e som, a digitalização do sinal revoluciona o sistema televisivo, disponibilizando via difusão conteúdos em novos formatos e suportando o funcionamento de um mecanismo de interatividade que transforma o telespectador em usuário ativo. O Brasil estabeleceu recentemente o seu padrão ISDB-Tb, e a televisão digital apresenta-se como uma tecnologia ainda pouco explorada e com considerável potencial de negócios. O middleware do sistema brasileiro, batizado como Ginga, permite o desenvolvimento de aplicações em linguagens declarativas ou procedurais. Este trabalho visa relatar o desenvolvimento de um protótipo em Java baseado no subsistema procedural do middleware Ginga, explorando uso do canal de retorno.*

Palavras-chaves: *TV Digital Interativa; Sistema Brasileiro de Televisão Digital; Middleware Ginga; Java; OpenGinga.*

1. Introdução

Embora não seja novidade em países mais desenvolvidos, a TV digital aberta vem ganhando destaque recentemente nas nações em desenvolvimento. Neste cenário, o Brasil se destaca pela criação de um quarto padrão aberto, juntando-se aos padrões americano, europeu e japonês e oferecendo um sistema com características consideradas como de vanguarda pelas instituições internacionais, gerando o interesse por parte de países da América Latina e África [Peduzzi, 2010; Soares, 2008].

O Governo Federal, mentor do Sistema Brasileiro de Televisão Digital, enxerga a TV digital aberta como instrumento de inclusão social e digital da grande parcela da população que não dispõe de recursos para usufruir de outros veículos. Este potencial é sustentado pela interatividade oferecida pela TV digital, sob a forma de aplicativos específicos, oferecendo uma gama de serviços interativos, como acesso a conteúdos de hipermídia, comércio eletrônico pela televisão, Internet e outros mais [Bastos, 2008].

A interatividade, característica que revoluciona a televisão não só para o telespectador, mas também sob o ponto de vista de cadeia de negócios, é possível graças à digitalização do conteúdo e transmissão, combinada com o funcionamento de componentes e mecanismos específicos, de modo especial o *middleware* - componente

presente nos receptores e nos ambientes de desenvolvimento de aplicações – e a inclusão de uma rede ou canal de retorno entre o receptor e o transmissor.

Para os desenvolvedores, o *middleware* é a base de trabalho no desenvolvimento de aplicações interativas para a TV digital, podendo ser em linguagens de paradigma declarativo, procedural ou híbridas, respeitando-se as linguagens adotadas pelos middlewares de cada padrão de TV digital. O *middleware* do padrão brasileiro é o Ginga.

A televisão digital no Brasil é considerada como uma inovação tecnológica ainda pouco explorada, limitando-se praticamente às equipes envolvidas nas pesquisas em prol do desenvolvimento do sistema. Entretanto, devido ao potencial que oferece e aliado às características do seu *middleware*, mostra-se como campo promissor. Embora uma parte do *middleware* ainda esteja em fase de implementação, alguns trabalhos já vêm sendo desenvolvidos servindo como referência às possibilidades oferecidas, de modo especial aqueles explorando o subsistema procedural Ginga-J.

2. Televisão Digital interativa

Além dos aspectos relacionados à qualidade de áudio e vídeo, a TV Digital oferece melhorias significativas em outros aspectos, como possibilitar a interoperabilidade com outras tecnologias – pelo fato de ser codificada digitalmente – e oferecer a possibilidade de interatividade para o telespectador. A TV digital caracteriza-se pela convergência tecnológica entre sistema televisivo, computador e redes de telecomunicação [Carvalho, 2008].

Resumidamente, podemos listar alguns dos benefícios, como compactação do sinal transmitido, transmissão mais robusta e menos suscetível a interferências, envio de fluxos múltiplos e simultâneos de áudio, vídeo e dados, recepção por dispositivos móveis e portáteis, além dos novos produtos e serviços que proporcionam interatividade [Breunig; Marques, 2008; Carvalho, 2008; Kneipp, 2009; Montez; Becker, 2005].

Os sistemas pioneiros da TV digital aberta são o americano e europeu, implantados na década de 90, e o japonês, em funcionamento via terrestre desde 2003 e serviu como modelo para o desenvolvimento do padrão brasileiro, denominado ISDB-Tb [Becker; Montez, 2004; Castro; Codato, 2008; Darós, 2004; Lopes, 2007].

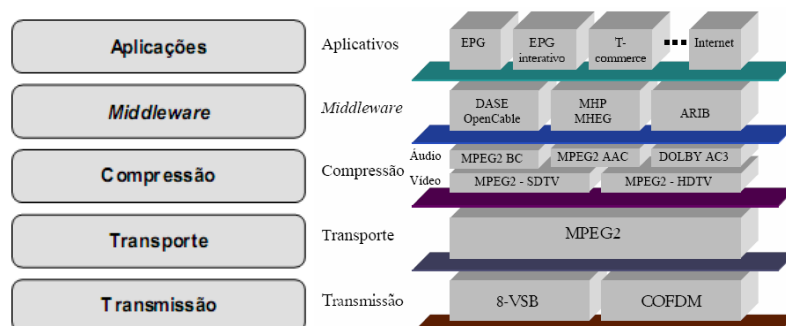


Figura 1. Arquitetura de camadas de um sistema de TV Digital e tecnologias envolvidas

Genericamente, a estrutura e funcionamento de um sistema de TV digital aberta pode ser representado por uma arquitetura de camadas semelhante ao conceito OSI de

redes, onde o conteúdo flui de camada em camada até chegar ao destino, sofrendo operações sucessivas, conforme exibido na Figura 1.

3. A interatividade

Baseado em estratégias comerciais, muito se vende hoje como dotado de interatividade, quando muitas vezes na realidade o que se vê é a presença de interação, e não de interatividade. Segundo Montez e Becker (2005) a interação é inerente ao ser humano e pode ocorrer diretamente entre dois ou mais entes, ao passo que a interatividade exige a presença meio eletrônico intermediário entre as partes envolvidas.

A interatividade na televisão representa uma quebra de paradigma. O usuário deixa de ser passivo, passando a interferir de fato na produção ou exibição dos conteúdos de várias formas, como escolha de ângulos de transmissão, solicitação de aplicações ou programas específicos, intervenção na grade de programação das emissoras e, em um grau mais alto de interatividade, atuando como produtor, enviando ao emissor conteúdo de sua própria autoria [Castro; Codato, 2008; Monteiro, 2002; Montez; Becker, 2005; Paschoal Neto; Carvalho, 2008].

Na TV digital, a interatividade pode ocorrer em vários níveis, havendo mais de uma classificação, entre elas aquela em níveis sugerida por Lemos (1997) e redefinida por Montez e Becker (2005), além daquela proposta pelo CNPq, tomando como referência a presença ou não do canal de retorno [Dall' Antonia, 2004; Tavares et al, 2007].

Os mecanismos essenciais para o funcionamento da interatividade são: a etapa de multiplexação e o *datacasting* via carrossel, a presença da camada de *middleware* como suporte ao desenvolvimento e execução das aplicações e o estabelecimento de um canal de retorno via rede de telecomunicação – Internet - entre o ambiente do receptor e o do transmissor (emissora) ou provedores de conteúdos e serviços da Internet.

4. Subsistema procedural Ginga-J e aplicações Xlet

O *middleware* é uma camada de software posicionada entre as aplicações e o sistema operacional/*hardware*, abstraindo detalhes de nível mais baixo e heterogeneidade de plataformas, facilitando o processo de desenvolvimento e distribuição de aplicações. Embora seja um conceito herdado dos sistemas computacionais, este componente se faz presente na arquitetura da TV digital e é fundamental no desenvolvimento das aplicações [Duarte, 2009; Etcharte, 2008; Montez; Becker, 2005; Soares, 2008].

No padrão brasileiro ISDB-Tb o *middleware* desenvolvido pelas instituições de pesquisa envolvidas é denominado Ginga. Compõe-se de um núcleo comum ou *Ginga Core*, do subsistema declarativo denominado *Ginga-NCL* e do subsistema procedural ou *Ginga-J*, além de uma ponte entre estes dois, permitindo o desenvolvimento e execução de aplicações híbridas – Figura 2 [Ginga, 2008].

A linguagem procedural adotada pelo *Ginga* é Java, assim como nos demais *middlewares* para TV digital aberta [Kulesza, 2009]. O subsistema *Ginga-J* foi concebido pelo Laboratório LAViD, da Universidade Federal da Paraíba. A composição inicial continha, além das bibliotecas de inovações voltadas ao contexto brasileiro, bibliotecas herdadas de outros *middlewares* e que exigiam o pagamento de *royalties*, vindo de encontro aos requisitos definidos pelo governo quando da implantação do

Sistema. Assim, em 2008 firmou-se uma parceria com a Sun *Microsystems* para a definição de uma especificação livre de *royalties* e aberta, substituindo os pacotes anteriores [Boquimpani, 2009; Java DTV, 2009; Zancanaro; Santos; Todesco, 2009].

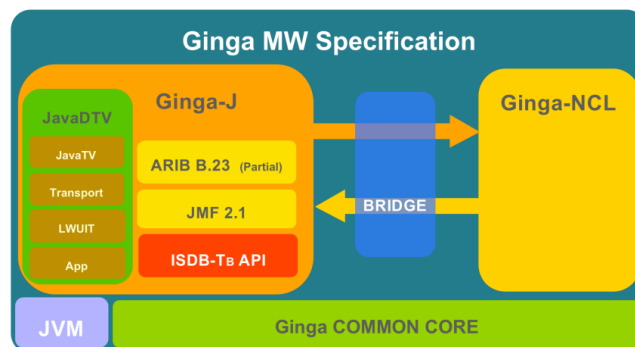


Figura 2. Estrutura do Ginga-J após a substituição dos pacotes GEM por JavaDTV

A especificação JavaDTV – Figura 2 - foi entregue e está atualmente em fase de implementação e substituição das bibliotecas antecessoras, a cargo do projeto Ginga CDN, vinculado ao laboratório LAViD e empenhado na construção de uma implementação de referência para o Ginga-J, denominada *OpenGinga*. O subsistema Ginga-J caracteriza-se por oferecer ao desenvolvedor o acesso a uma série de recursos avançados, além de permitindo um maior controle sobre o comportamento da aplicação desenvolvida, característica das linguagens procedurais [Boquimpani, 2009; Duarte, 2009; Etcharte, 2008; Kulesza; Ferreira, 2009; Openginga, 2010].

As aplicações desenvolvidas para TV Digital no ambiente procedural são denominadas *Xlets*, de construção, comportamento e ciclo de vida semelhantes aos *Applets Web* e *MIDlets* dos dispositivos móveis. Normalmente uma aplicação *Xlet* consiste de uma classe que implementa a interface *javax.tv.xlet.Xlet*, da biblioteca Java TV [Abnt, 2007; Abnt, 2010; Becker et al, 2005; Kulesza, 2009].

5. Desenvolvimento de um protótipo interativo baseado em Ginga-J

O protótipo proposto foi desenvolvido utilizando-se o emulador *GingaJ-Emulator*, integrante do projeto *OpenGinga*, conduzido pela equipe do Ginga CDN e que visa o desenvolvimento de uma implementação de referência do subsistema procedural do *middleware* brasileiro.

Optou-se pelo desenvolvimento de uma aplicação do tipo *T-Commerce*, sob a forma de uma loja virtual de artigos esportivos. Utilizando o controle remoto como dispositivo de interação, o usuário pode navegar entre itens arranjados em categorias, visualizando detalhes e selecionando itens para compra.

A arquitetura do sistema é composta por um *Xlet* cliente (o protótipo propriamente dito) que é executado no ambiente emulador e receberá a interação do usuário, e por um *WebService* remoto. O conteúdo exibido ao usuário é construído com base nos dados retornados de requisições feitas ao servidor remoto via rede de telecomunicação ou Internet, configurando assim um ambiente de interatividade remota, caracterizada pela dependência da existência de um canal de retorno entre o cliente e o servidor.

O *Xlet* foi desenvolvido e executado no ambiente fornecido pelo *GingaJ-Emulator*. A ferramenta é baixada do repositório do projeto Ginga CDN e importada para a IDE Eclipse como um projeto, e após ser configurada e compilada, fornece não só as bibliotecas necessárias ao desenvolvimento de aplicações – baseadas no conjunto do subsistema Ginga-J - como também um ambiente emulador de aparelho receptor, que permite registrar, executar e testar as aplicações desenvolvidas, fornecendo um ecrã e controle remoto virtuais.

A plataforma base, tanto do emulador como da implementação de referência OpenGinga é a Java ME - Java *Connected Device Configuration framework* (CDC) 1.1, *Foundation Profile* (FP) 1.1 e *Personal Basis Profile* (PBP) 1.1 [Boquimpani, 2009; Kulesza; Ferreira, 2009]. A Figura 3 ilustra a execução do protótipo desenvolvido no emulador, a partir da IDE Eclipse:

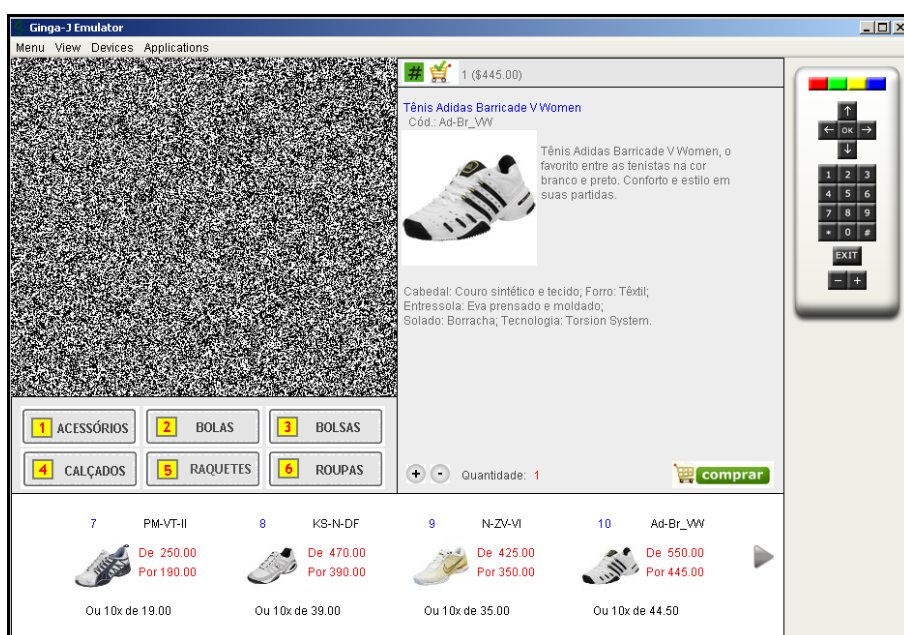


Figura 3. Exibição de Calçados no painel inferior, e detalhes de um item selecionado

A interface gráfica foi construída com base nos componentes da API LWUIT, incluída na implementação JavaDTV entregue pela Sun ao Fórum SBTVD. Além dos pacotes gráficos, a especificação inclui uma série de API's – entre elas Java TV – que fornecem funcionalidades como acesso a componentes de nível mais baixo, transação segura de dados, suporte ao uso de dispositivos avançados, entre outros.

A comunicação feita com o servidor remoto – onde uma base de dados persiste as informações relativas aos artigos comercializados - é suportada pelas bibliotecas *java.net* e *java.io*, fornecidas pela plataforma Java ME. Os mecanismos empregados foram as requisições HTTP ao servidor via URL (*java.net*), utilizando-se fluxos *ObjectInputStream* e *ObjectOutputStream* (*java.io*) para o envio e recebimento de coleções de objetos entre o servidor e o cliente e vice-versa, via requisição estabelecida pelo canal de retorno.

Além da passagem de objetos via requisição, implantou-se um mecanismo de criptografia com chaves privadas e simétricas em algoritmo AES de 256 bits de posição, encriptando/decriptando não só sobre parâmetros *String*, mas também objetos.

Como a implementação da especificação ainda está em andamento, algumas bibliotecas ainda não estão totalmente concluídas, de modo especial aquelas relacionadas à interface gráfica e eventos de interação – API LWUIT. Por isso, dificuldades foram enfrentadas em relação à captura de eventos e exibição de conteúdo, dificultando a implantação do mecanismo de interação usuário-aplicação.

A interface foi construída com base em dois componentes gráficos: *Frame*, que fornece o contêiner principal da interface e o *widget Label*, cuja única funcionalidade é exibir conteúdo textual ou imagem. Essa limitação de componentes exigiu a implantação de um mecanismo baseado no uso dos botões coloridos do controle remoto como gatilhos de eventos, associados a estados da aplicação, representados por variáveis booleanas operando em sistema de exclusão mútua (semáforo), como forma de suprir a funcionalidade dos componentes ainda não disponíveis, como botões, campos de texto, diálogos e outros.

6. Trabalhos correlatos

A literatura apresenta alguns trabalhos envolvendo aplicações e protótipos baseados na plataforma Java para TV digital, em linhas de estudo semelhantes à desta pesquisa. Darós (2004) descreve o desenvolvimento de um protótipo de *T-Commerce* baseado no *middleware* MHP, do padrão europeu.

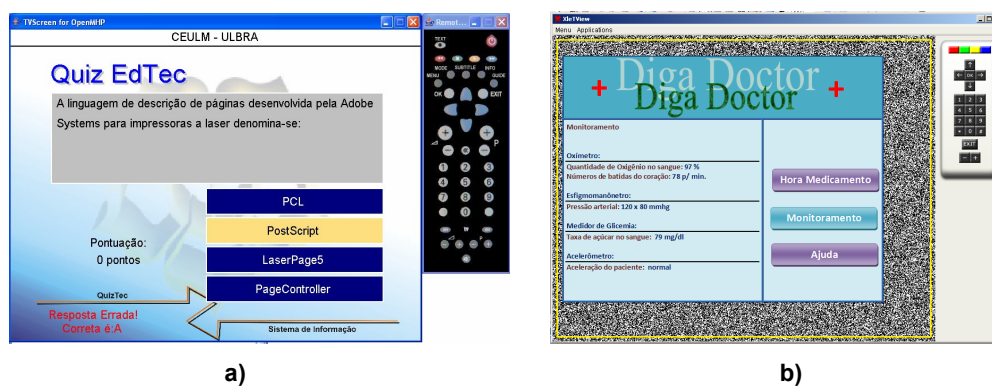


Figura 4. a) Protótipo “Quiz Edtec”; b) protótipo “Diga Doctor”

Pinto, Queiroz-Neto e Silva (2007) descrevem o desenvolvimento de um protótipo de jogo educacional para TV Digital, denominado “Quiz Edtec”, construído com base na especificação anterior do subsistema Ginga-J, que continha as bibliotecas herdadas do GEM (Havi e DAVIC) substituídas posteriormente pela especificação aberta JavaDTV – Figura 4 “a”.

Silva Santos et al (2009) apresentam o protótipo DIGA *Doctor*, módulo demonstrativo do projeto DIGA GINGA, idealizado pela Universidade Federal do Ceará em um projeto destinado a utilizar a estrutura do *set-top box* e *middleware* Ginga como ambiente de desenvolvimento de aplicações voltadas ao cidadão em sua residência. Trata-se da idealização de um projeto a ser implementado – Figura 4 “b”.

Diferentemente dos trabalhos acima relacionados, o protótipo proposto foi desenvolvido com base na especificação JavaDTV do subsistema Ginga-J, construída para substituir a funcionalidade da especificação anterior, passível de cobrança de *royalties*. Embora a implementação da especificação ainda não esteja concluída no

momento, o uso do *OpenGinga/GingaJ-Emulator* já possibilita a construção de protótipos interativos que explorem as funcionalidades oferecidas pelo Ginga-J.

7. Conclusão

O protótipo desenvolvido – ápice dos trabalhos realizados durante a pesquisa – serve como demonstrativo prático das múltiplas possibilidades decorrentes da utilização da especificação Ginga no desenvolvimento de aplicações interativas para a TV Digital.

Em particular as bibliotecas do subsistema procedural Ginga-J, objeto específico deste estudo, que apesar das dificuldades enfrentadas e limitações encontradas- em virtude de que boa parte da especificação ainda esteja em trabalho de implantação ou implementação – possibilitou o desenvolvimento de um protótipo com interface gráfica consideravelmente rica e funcional, com interatividade local satisfatória e na medida do possível intuitiva. Esses problemas serão resolvidos gradativamente, na medida em que as classes da API LWUIT forem implementadas pela equipe de desenvolvimento da implementação de referência *OpenGinga*, que vem se mostrando como um instrumento importante na disseminação da tecnologia junto à comunidade de desenvolvedores brasileiros, de modo especial os que se dedicam à plataforma e linguagem Java.

No tocante à questão da interatividade no âmbito do protótipo desenvolvido, os recursos de rede oferecidos pela especificação Ginga-J viabilizaram o estabelecimento de um canal de interatividade remota e dedicada via canal de retorno (Internet) de maneira simples, tal qual em aplicações *web*. Nesse contexto, a presença de bibliotecas como *java.net* e *java.io* merecem destaque.

Considerando o contexto brasileiro e a questão da interatividade, é evidente a necessidade de se estabelecer políticas que permitam a criação de uma estrutura de comunicações que alcance as classes mais baixas da sociedade. Sem o emprego de um canal de retorno via Internet, as possibilidades de interatividade na TV Digital são limitadas; logo, a questão de se oferecer acesso de qualidade e baixo custo à população carente precisa ser acompanhada, não só pelo setor público, mas pela sociedade brasileira em um conjunto de esforços.

Referências

ABNT. Associação Brasileira de Normas Técnicas. (2010). **NBR 15606-4:2010**: Ginga-J – Ambiente para a execução de aplicações procedurais. 91 f. Primeira edição (válida a partir de 13.05.2010), Brasília, 2010. Disponível em: <http://www.dtv.org.br/download/pt-br/ABNTNBR15606-4_2010Ed1.pdf>. Acesso em 10 mai. 2010.

_____. Associação Brasileira de Normas Técnicas. (2007). **PROJETO 00:001.85-006/4**: Ginga-J: Ambiente para a execução de aplicações procedurais. 36 f. Primeiro projeto, Brasília, set. 2007. Disponível em: <<http://lince.dc.ufscar.br/home/pos-graduacao/Material%20TVD/parte4-gingaJ.pdf>>. Acesso em 15 out. 2009.

BASTOS, P. (2008). A TV Digital e as Novas Possibilidades de Interatividade. IN: XXXI Congresso Brasileiro de Ciências da Comunicação, Natal, 2008. **Anais...** Natal: Intercom, 2008. Disponível em: <<http://www.intercom.org.br/papers/nacionais/2008/resumos/R3-0768-1.pdf>>. Acesso em: 15 out. 2009.

BECKER, V. et al. (2005). Datacasting e Desenvolvimento de Serviços e Aplicações para TV Digital Interativa. In: TEIXEIRA, Cesar Augusto Camilo; BARRÈRE, Eduardo; ABRÃO, Iran Calixto. (Org.). **Web e Multimídia: Desafios e Soluções**. Poços de Caldas, 2005, v. 01, p. 01-30. Disponível em: <<http://www.itvproducoesinterativas.com.br/pdfs/A-Datacasting-webmidia.pdf>>. Acesso em 10 out. 2009.

BECKER, V.; MONTEZ, C.(2004). TV Digital Interativa: conceitos e tecnologias. IN: Minicursos WebMedia & LA-Web, 2004, Ribeirão Preto. **Anais...** Ribeirão Preto: WebMedia, 2004. Disponível em: <<http://www.itvproducoesinterativas.com.br/pdfs/A-TV-Digital-Interativa-Conceitos-Tecnologias.pdf>>. Acesso em: 05 mar. 2010.

BOQUIMPANI, A. (2009). Explorando o Ginga-J: como atingir cenários reais de interatividade na TV. IN: JustJava, 2009, São Paulo. **Anais...** São Paulo:SUCESU-SP, 2009. Disponível em: <<http://www.sucesusp.org.br/mailling2009/congresso/justjava/apresentacoes/AuditorioPrincipal/TVDigital.pdf>>. Acesso em: 2 nov. 2009.

BREUNIG, M.; MARQUES, V. (2008). A digitalização da TV. IN: XXXI Congresso Brasileiro de Ciências da Comunicação – Natal. **Anais...** Natal, 2008. Disponível em: <<http://www.intercom.org.br/papers/nacionais/2008/resumos/R3-1569-1.pdf>>. Acesso em: 15 out. 2009.

CARVALHO, E. (2008). **Uma plataforma modular para testes com interatividade na TV Digital brasileira**. 2008. 119 f. Dissertação (Mestrado em Engenharia). Área de concentração: Sistemas Eletrônicos – USP: Escola Politécnica, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-30052008-110812/>>. Acesso em: 5 out. 2009.

CASTRO, J.; CODATO, H. (2008). Sistema Brasileiro de Televisão Digital: marketing, tendências e aplicabilidade do audiovisual. IN: 7º Encontro Internacional de Arte e Tecnologia, 2008, Brasília. **Anais...** Brasília: #7 ART, 2008 Disponível em: <http://www.jobertcastro.com/artigo_sbtvd.pdf>. Acesso em: 10 out. 2009.

DALL'ANTONIA, J. (2004). **Cadeia de Valor FUNTTEL – Projeto Sistema Brasileiro de TV Digital**. OS 40539. Versão PD 30.12.36^a.002A/RT-02-AB, 95 f. Campinas, CPqD, 2004. (Relatório Técnico). Disponível em: http://sbtvd.cpqd.com.br/cmp_tvdigital/divulgacao/anexos/26_42_cadeia_de_valo.pdf. Acesso em 16 out. 2009.

DARÓS, J.(2004). **Interfaces de desenvolvimento de aplicações para TV Digital baseado no middleware MHP**. 2004. 53 f. Trabalho de Conclusão de Curso (Bacharelado) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, FURB, Blumenau, 2004. Disponível em: <<http://campeche.inf.furb.br/tccs/2004-II/2004-2joeladarosap.pdf>>. Acesso em: 18 out. 2009.

DUARTE, D. (2009). **Desenvolvimento de um protótipo de interatividade para TV Digital através do middleware Ginga**. 2009. 61 f. Trabalho de Conclusão de Curso (Bacharelado) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2009.

ETCHARTE, V. (2008). **Receptores e canal de interatividade para o Sistema Brasileiro de TV Digital**. 2008. Artigo apresentado à disciplina de Fundamentos de Sistemas Multimídia, Escola de Engenharia, Universidade Federal Fluminense, Niterói, 2008. Disponível em: < <http://www.midiacom.uff.br/~debora/fsmm/trab-2008-2/receptor.pdf>>. Acesso em: 06 mar. 2010.

GINGA Digital TV Middleware Specification. (2008). **TV Interativa se faz com Ginga**. Página de apresentação do middleware Ginga, 2008. Disponível em: <<http://ginga.org.br>>. Acesso em: 01 out. 2009.

JAVA DTV para Ginga-J sem royalties. (2009). **Produção Profissional**, São Paulo, v. 1, n.85, jan. 2009. Disponível em: <<http://www.producaoprofissional.com.br/files/editions/ed1236005099.pdf>>. Acesso em: 4 out. 2009.

KNEIPP, V. (2009). Trajetória da TV Digital no Ceará e a interatividade do Ginga. IN: 1º Simpósio Internacional de Televisão Digital (SIMTVD), 2009, Bauru. **Anais...** Bauru: SIMTVD, 2009. Disponível em: <[http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/KNEIPP%20-%20Trajet%3ria%20da%20TV%20Digita%20no%20Cear%e1%20\(500-511\).pdf](http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/KNEIPP%20-%20Trajet%3ria%20da%20TV%20Digita%20no%20Cear%e1%20(500-511).pdf)>. Acesso em: 02 mar. 2010.

KULEZSA, R. (2009). **API JavaTV: criando e controlando aplicações Java no Ginga**. Overmedia Networks – Treinamento Ginga Módulo 1, 2009. Disponível em: <<http://graphs.ucpel.tche.br/tvdi/OficinaGingaJ-Parte2.pdf>>. Acesso em 02 mar. 2010.

KULEZSA, R.; FERREIRA, J. (2009). Desenvolvimento Ginga-J - Java DTV- OpenGinga. IN: Fórum Internacional de Software Livre: Oficina Ginga-J, 2009, Porto Alegre. **Anais...** Porto Alegre: FISL, 2009. Disponível em: <<http://graphs.ucpel.tche.br/tvdi/OficinaGingaJ-Parte1.pdf>>. Acesso em: 15 out. 2009.

LEMOS, A. (1997). Anjos interativos e retribalização do mundo: sobre interatividade e interfaces digitais. **Tendências XXI**, Lisboa, n. 2, p. 19-29, set. 1997. Disponível em: < <http://www.facom.ufba.br/ciberpesquisa/lemos/interac.html>>. Acesso em: 15 mar. 2010.

LOPES, D. (2007). Sistema Brasileiro de TV Digital: caminhos percorridos e implantação. IN: V Congresso Nacional de História na Mídia, 2007, São Paulo. **Anais...** São Paulo: Intercom, 2007. Disponível em: <<http://www.rp-bahia.com.br/biblioteca/hist-midia2005/resumos/R0097-1.pdf>>. Acesso em: 09 out. 2009.

MONTEIRO, M.(2002). **TV interativa e seus caminhos**. 2002. 84 f. Dissertação (Mestrado em Computação). Área de concentração: Engenharia de Computação – Instituto de Computação, Universidade Estadual de Campinas, Campinas. Disponível em: < <http://libdigi.unicamp.br/document/?code=vtls000294986>> Acesso em: 10 abr. 2010.

MONTEZ, C.; BECKER, V. (2005). **TV Digital interativa: conceitos, desafios e perspectivas para o Brasil**. 2. ed. ampliada e revisada. Florianópolis: da UFSC, 2005. 201 p. Disponível em: <http://www.labtvd.com.br/novo/attachments/061_TV-Digital-Interativa_2a_EDICAO.pdf>. Acesso em: 10 mar. 2010.

OPENGINGA. **OpenGinga** - Implementação de referência do middleware brasileiro de TV Digital. (2010). Disponível em: <<http://www.openginga.org/>>. Disponível em cache: <<http://webcache.googleusercontent.com/search?q=cache:S8i2buSmMPgJ:www.openginga.org/about.html+O+que+%C3%A9+OpenGinga+%3F&cd=1&hl=pt-BR&ct=clnk&gl=br>>. Acesso em: 01 mar. 2010.

PASCHOAL NETO, J.; CARVALHO, J.(2008). Os novos paradigmas da informação e a TV digital: o papel das TVs universitárias na construção de conteúdos de maneira colaborativa através de redes interdisciplinares . **Informação & Sociedade : Estudos**, João Pessoa, v.18, n.3, p. 59-72, set/dez. 2008. Disponível em: <<http://www.ies.ufpb.br/ojs2/index.php/ies/article/view/1806/2272>>. Acesso em: 6 out. 2009.

PEDUZZI, P.(2010). Ministros africanos manifestam preferência pelo padrão de TV digital nipo-brasileiro. **Agência Brasil**: Empresa Brasil de Comunicação, Brasília, 12 ju. 2010. Disponível em: <http://www.direitoacomunicacao.org.br/content.php?option=com_content&task=view&id=6872>. Acesso em: 01 dez. 2010.

PINTO, L.; QUEIROZ-NETO, J.; SILVA, K.(2007). Tecnologias para o desenvolvimento de aplicações educacionais para TV Digital. IN: II Congresso de Pesquisa e Inovação da Rede Nordeste de Educação Tecnológica, 2007, João Pessoa. **Anais...** João Pessoa: CONNEPI, 2007. Disponível em: <http://www.redenet.edu.br/publicacoes/arquivos/20080212_081200_INFO-078.pdf>. Acesso em: 12 out. 2009.

SILVA SANTOS, M. et al. (2009). Desenvolvendo aplicações Home Care no ambiente Ginga. IN: 35ª InfoBrasil TI & Telecom e II Congresso Tecnológico InfoBrasil, 2009, Fortaleza. **Anais...** Fortaleza: InfoBrasil, 2009. Disponível em: <<http://www.infobrasil.inf.br/iConstructor/Custom/anais2009/Desenvolvendo%20Aplicacao%20C3%A7%C3%B5es%20Home%20Care%20no%20Ambiente%20GINGA.pdf>>. Acesso em: 12 out. 2009.

SOARES, L.(2008) As múltiplas possibilidades do middleware Ginga. **Produção Profissional**: Revista de comunicação e técnica audiovisual, São Paulo, p. 76-83, jun. 2008. Disponível em: <ftp://ftp.telemidia.puc-rio.br/~lfgs/docs/journalpapers/2008_06_soares.pdf>. Acesso em: 7 out. 2009.

TAVARES, T.et al. (2007). A TV Digital interativa como ferramenta de apoio à educação infantil. **Revista Brasileira de Informática na Educação**, Porto Alegre, Vol. 15, n. 2, mai./ago. 2007. Disponível em: <<http://bibliotecadigital.sbc.org.br/download.php?paper=939>>. Acesso em: 10 out. 2009.

ZANCANARO, A.; SANTOS, P.; TODESCO, J. (2009). Ginga-J ou Ginga-NCL: características das linguagens de desenvolvimento de recursos interativos para a TV Digital. IN: 1º Simpósio Internacional de Televisão Digital (SIMTVD), 2009, Bauru. **Anais...** Bauru: UNESP, 2009. Disponível em: <<http://www2.faac.unesp.br/pesquisa/lecotec/eventos/simtvd/anais/ZANCANARO;%20SANTOS;%20TODESCO%20-%20GINGA-J%20OU%20GINGA-NCL%20%281084-1108%29.pdf>>. Acesso em: 15 mar. 2010.