



**CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB**

**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**PROJETO FINAL**

**WANESSA DE SOUSA BASTOS**

**SOLUÇÃO DE INTERATIVIDADE PARA TV DIGITAL EM  
PADRÃO ISDB-T<sub>B</sub> UTILIZANDO CANAL DE RETORNO VIA INTERNET**

**Orientador: Prof. Francisco Javier de Obaldia Diaz**

Brasília  
Junho/2010

**WANESSA DE SOUSA BASTOS**

**SOLUÇÃO DE INTERATIVIDADE PARA TV DIGITAL EM  
PADRÃO ISDB-T<sub>B</sub> UTILIZANDO CANAL DE RETORNO VIA INTERNET**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientador: Prof. Francisco Javier  
de Obaldia Diaz

Brasília  
Junho/2010

**WANEISSA DE SOUSA BASTOS**

**SOLUÇÃO DE INTERATIVIDADE PARA TV DIGITAL EM  
PADRÃO ISDB-T<sub>B</sub> UTILIZANDO CANAL DE RETORNO VIA INTERNET**

Trabalho apresentado ao Centro  
Universitário de Brasília  
(UniCEUB) como pré-requisito  
para a obtenção de Certificado de  
Conclusão de Curso de Engenharia  
de Computação.

Orientador: Prof. Francisco Javier  
de Obaldia Diaz

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,  
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -  
FATECS.

---

Prof. Abiezer Amarilia Fernandez  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Francisco Javier de Obaldia Diaz – Mestre em Engenharia Elétrica  
Orientador

---

Prof. Flávio Antônio Klein - Mestre em Estatística e Métodos Quantitativos  
UniCEUB

---

Prof. João Marcos Souza Costa - Especialista em Matemática  
UniCEUB

---

Prof. Marco Antônio de Oliveira Araújo - Mestre em Ciência da Computação  
UniCEUB

## DEDICATÓRIA

Dedico este trabalho a minha filha Ana Beatriz, que em meio a tantos acontecimentos, foi quem me fez superar e acreditar que não devemos nunca desistir.

O dom da maternidade é a maior benção que Deus poderia ter me dado.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado o dom da existência e condições de estar aqui, finalizando esse curso.

Agradeço a meus pais, Edson e Helene, por sempre me apoiarem em todos os momentos da minha vida. Sejam bons ou ruins, eles sempre estiveram ao meu lado com todo amor, carinho e apoio que um filho merece ter.

Aos meus irmãos, Wagner e Waleria, que juntos me apoiaram em todos os momentos difíceis com muita dedicação, amor e carinho.

A todos os meus amigos que me apoiaram e me ajudaram não apenas nesse projeto, mas em todas as dificuldades passadas. Em especial a Daniela Santos e Sávio Bueno.

Aos meus novos amigos que, graças a este projeto, conheci e obtive muita ajuda: Marco Munhoz e Maurício Perdomo.

Ao UniCEUB e todos os professores do curso de Engenharia da Computação que com muita dedicação me mostraram o melhor caminho a seguir. Em especial ao coordenador Abiézer e meu orientador, que muito me ajudou neste processo, professor Javier.

A todos, muito obrigada.

## SUMÁRIO

<b>DEDICATÓRIA .....</b>	<b>3</b>
<b>AGRADECIMENTOS.....</b>	<b>4</b>
<b>LISTA DE FIGURAS.....</b>	<b>8</b>
<b>LISTA DE TABELAS.....</b>	<b>10</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>11</b>
<b>RESUMO.....</b>	<b>13</b>
<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>15</b>
<b>1.1. Motivação.....</b>	<b>16</b>
<b>1.2. Objetivo Geral .....</b>	<b>16</b>
<b>1.3. Objetivos Específicos .....</b>	<b>17</b>
<b>1.4. Metodologia Científica.....</b>	<b>17</b>
<b>1.5. Estrutura do Trabalho .....</b>	<b>18</b>
<b>1.6. Resultados Esperados .....</b>	<b>18</b>
<b>CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA .....</b>	<b>19</b>
<b>2.1. A Implantação da TV Digital no Brasil.....</b>	<b>19</b>
<b>2.2. O Ginga.....</b>	<b>21</b>
<b>2.3. O canal de Interatividade .....</b>	<b>23</b>
<b>2.4. Software Livre .....</b>	<b>25</b>
<b>2.5. Proposta do Trabalho.....</b>	<b>25</b>
<b>CAPÍTULO 3 - REFERENCIAL TEÓRICO .....</b>	<b>27</b>
<b>3.1. Interatividade .....</b>	<b>27</b>
<b>3.2. Bidirecionalidade.....</b>	<b>30</b>
<b>3.3. A Linguagem NCL.....</b>	<b>31</b>
<b>3.4. A Linguagem Lua.....</b>	<b>45</b>
<b>3.5 Hipertexto .....</b>	<b>45</b>
<b>3.6. Hiperídia.....</b>	<b>46</b>
<b>3.7. Transmissão Analógica X Digital .....</b>	<b>47</b>
3.7.1 Digitalização.....	47
3.7.2 Analógico X Digital .....	48
<b>3.8. Imagem e Padrões de cor.....</b>	<b>54</b>
<b>3.9. Áudio.....</b>	<b>58</b>
<b>3.10. Datacasting.....</b>	<b>59</b>
<b>3.11. Recepção Móvel (1-SEG) .....</b>	<b>59</b>

<b>3.12. Arquitetura .....</b>	<b>60</b>
<b>3.13. TV Digital no Mundo.....</b>	<b>62</b>
3.13.1 ATSC – Padrão Americano.....	62
3.13.1.1. Middleware – Padrão Americano .....	63
3.13.2 DVB – Padrão Europeu .....	64
3.13.2.1. Middleware – Padrão Europeu .....	65
3.13.3 ISDB – Padrão Japonês.....	66
3.13.3.1. Middleware – Padrão Japonês .....	66
<b>3.14. TV Digital no Brasil.....</b>	<b>70</b>
<b>3.15. Sistema Brasileiro de TV Digital (SBTVD).....</b>	<b>71</b>
3.15.1 Características e especificações técnicas.....	72
3.15.1.1 Mobilidade e Portabilidade .....	73
3.15.1.2 Multiprogramação .....	74
3.15.1.3 A TV Interativa .....	75
3.15.1.4 Codificação de áudio .....	77
3.15.1.5 Codificação de vídeo .....	79
3.15.1.6 Transmissão .....	84
3.15.1.6.1. Seção de Codificação de Canal.....	88
3.15.1.6.2. Modulação de bits.....	90
3.15.1.6.3. Multiplexador (MUX).....	93
3.15.1.6.4. Modulações em DVB-S2 .....	96
3.15.1.7 Carrossel de objetos.....	97
<b>3.16 Middleware Ginga .....</b>	<b>99</b>
3.16.1 Arquitetura Ginga.....	100
3.16.1.1 Ginga-NCL .....	101
3.16.1.2 Ginga-J.....	102
3.16.1.3 Ginga-CC.....	103
<b>3.17 Segurança.....</b>	<b>104</b>
<b>3.18 O Canal de Retorno .....</b>	<b>105</b>
<b>3.19 Normas ABNT .....</b>	<b>106</b>
<b><i>CAPÍTULO 4 - PROPOSTA DE SOLUÇÃO .....</i></b>	<b><i>109</i></b>
<b>4.1. Arquitetura do código .....</b>	<b>110</b>
<b>4.2 O Código .....</b>	<b>112</b>
4.2.1 Documento NCL .....	112
4.2.2 Regiões .....	113
4.2.3 Descritores .....	114
4.2.4. Conectores .....	114
4.2.5 Nós de mídia .....	115
4.2.6 Links.....	115
<b>4.3. Servidor Web – infotelbsb.com.br.....</b>	<b>115</b>
<b><i>CAPÍTULO 5 – APLICAÇÃO E DEMONSTRAÇÃO DA SOLUÇÃO COM</i></b>	
<b><i>RESULTADOS .....</i></b>	<b><i>116</i></b>
<b>5.1. Instalação da Máquina Virtual .....</b>	<b>117</b>
<b>5.2. Execução e resultados da interatividade .....</b>	<b>120</b>

5.3. A comunicação entre máquinas.....	128
<b><i>CAPÍTULO 6 - CONCLUSÃO</i></b> .....	<b>129</b>
6.1 Sugestões Para Trabalhos Futuros.....	130
<b><i>REFERÊNCIA BIBLIOGRÁFICA</i></b> .....	<b>131</b>
<b><i>APÊNDICE A – CÓDIGO FONTE ENQUETE INTERATIVO</i></b> .....	<b>136</b>
<b><i>APÊNDICE B – CÓDIGO FONTE ARQUIVO CAUSALCONNBASE.NCL</i></b> .....	<b>149</b>
<b><i>APÊNDICE C – CÓDIGO FONTE ARQUIVOS LUA</i></b> .....	<b>156</b>
<b><i>APÊNDICE D – CÓDIGO FONTE ARQUIVOS PHP</i></b> .....	<b>181</b>



## LISTA DE FIGURAS

<i>Figura 01 – Cronograma TV Digital no Brasil (Fonte: Minicom, 2006).....</i>	<i>20</i>
<i>Figura 02 – Canal de Interatividade (Fonte: wirelessbrasil.org, 2010).....</i>	<i>24</i>
<i>Figura 03 – Estrutura de Regiões em um arquivo NCL .....</i>	<i>33</i>
<i>Figura 04 – Região do Programa NCL .....</i>	<i>33</i>
<i>Figura 05 – Atributos das regiões em um arquivo NCL (Fonte: dtvgames, wordpress, 2010).....</i>	<i>34</i>
<i>Figura 06 – Exemplificação do atributo zIndex NCL (Fonte: dtvgames, wordpress, 2010)....</i>	<i>35</i>
<i>Figura 07 – Estrutura dos descritores em um arquivo NCL (Fonte: adaptado,dtvgames, wordpress, 2010) .....</i>	<i>35</i>
<i>Figura 08 – Nós do programa NCL.....</i>	<i>38</i>
<i>Figura 09 – Nó de Contexto do programa NCL.....</i>	<i>40</i>
<i>Figura 10 – Porta do programa NCL.....</i>	<i>41</i>
<i>Figura 11 – Estrutura Básica de um arquivo NCL (Fonte: Tutorial NCL, 2007) .....</i>	<i>44</i>
<i>Figura 12 – Formas de apresentação de hipertextos (Fonte: DEE-LUCAS, 1996). .....</i>	<i>46</i>
<i>Figura 13 – Processo de Digitalização .....</i>	<i>48</i>
<i>Figura 14 – Divisão da Banda 19Mbps TV Digital.....</i>	<i>49</i>
<i>Figura 15 – Modulação ASK.....</i>	<i>50</i>
<i>Figura 16 – Modulação FSK.....</i>	<i>50</i>
<i>Figura 17 – Modulação 2-PSK .....</i>	<i>51</i>
<i>Figura 18 – Modulação 4-PSK .....</i>	<i>51</i>
<i>Figura 19 – Modulação 8-QAM.....</i>	<i>52</i>
<i>Figura 20 – Constelações 4-QAM e 8-QAM.....</i>	<i>53</i>
<i>Figura 21 – Variações das constelações 16-QAM.....</i>	<i>53</i>
<i>Figura 22 – Espectro Eletromagnético (Fonte: SBFísica,2010) .....</i>	<i>54</i>
<i>Figura 23 – Aspecto 4:3.....</i>	<i>55</i>
<i>Figura 24 – Aspecto 16:9.....</i>	<i>55</i>
<i>Figura 25 – HD x SD.....</i>	<i>56</i>
<i>Figura 26 – Padrões de cor no mundo (Fonte: Wikipédia, 2010) .....</i>	<i>57</i>
<i>Figura 27 – Arquitetura da TV Digital (Fonte: Grupo Ginga Goiás, 2010).....</i>	<i>61</i>
<i>Figura 28 - Padrões de Referência do SBTVD (Fonte: Soares e Barbosa, 2009).....</i>	<i>61</i>
<i>Figura 29 – Middleware Americano – DASE (Fonte: ATSC, 2009).....</i>	<i>64</i>
<i>Figura 30 – Middleware Europeu – DVB (Fonte: “TV Interactiva”, 2009).....</i>	<i>66</i>
<i>Figura 31 – Middleware Japonês – ARIB (Fonte: NHK, 2009) .....</i>	<i>67</i>
<i>Figura 32 – Mapa dos Sistemas de TV Digital no Mundo (Fonte: DTVStatus, 2010).....</i>	<i>68</i>
<i>Figura 33 – Estados com TV Digital (DTV,2010).....</i>	<i>71</i>
<i>Figura 34 – Segmento I-SEG (Fonte: Wordpress, Armando Moraes, 2008).....</i>	<i>73</i>
<i>Figura 35 – Primeiro programa interativo no mundo (Fonte: Teixeira, 2009) .....</i>	<i>75</i>
<i>Figura 36 – Faixa audível – ouvido humano (Fonte: Cristóvam, 2010).....</i>	<i>77</i>
<i>Figura 37 – Procedimentos de transmissão e codificação de áudio (Fonte: ABNT, 2008)....</i>	<i>78</i>
<i>Figura 38 – Codificação de diferenças – Frames I e Frames P .....</i>	<i>81</i>
<i>Figura 39 – Sequência com frames I, B e P (Fonte: Axis, 2008).....</i>	<i>82</i>
<i>Figura 40 – Representação esquemática do processo de codificação (Fonte: ABNT, 2007) .</i>	<i>83</i>
<i>Figura 41 – Fracionamento da sequência de vídeo em macroblocos (Fonte: ABNT, 2007) ..</i>	<i>84</i>

<i>Figura 42 – Bloco funcional do sistema de transmissão do SBTVD (Fonte: Produção Profissional, 2008)</i> .....	85
<i>Figura 43 – Configuração do SBTVD (Fonte: Produção Profissional, 2008)</i> .....	86
<i>Figura 44 – Exemplo de configuração de camadas (Fonte: Produção Profissional, 2008)</i> ...	86
<i>Figura 45 – Configuração básica do transmissor digital (Fonte: Produção Profissional, 2008)</i> .....	87
<i>Figura 46 – Seção de codificação de canal (Fonte: Produção Profissional, 2008)</i> .....	89
<i>Figura 47 – Diagrama de modulação 16-QAM (Fonte: Produção Profissional, 2008)</i> .....	90
<i>Figura 48 – Configuração para modulação das portadoras (Fonte: Produção Profissional, 2008)</i> .....	92
<i>Figura 49 – Fluxo de dados ou Transport Stream (TS) (Fonte: SET, 2009)</i> .....	93
<i>Figura 50 – Modulação DVB-S2 (Fonte: SET, 2009)</i> .....	97
<i>Figura 51 – Representação do carrossel</i> .....	99
<i>Figura 52 – Camada Middleware Ginga (Fonte: Soares e Barbosa, 2009)</i> .....	99
<i>Figura 53 – Arquitetura Ginga (Fonte: Produção Profissional, 2008)</i> .....	100
<i>Figura 54 – Estrutura do ambiente de aplicações (Fonte: Produção Profissional/2008)</i> ....	101
<i>Figura 55 – APIs Ginga-J (Fonte: GingaDF, 2008)</i> .....	102
<i>Figura 56 – Canal de interatividade bidirecional com acesso à internet (Fonte: ABNT, 2008)</i> .....	106
<i>Figura 57 – Diagrama do Projeto</i> .....	109
<i>Figura 58 – Visualização da Enquete Interativa</i> .....	110
<i>Figura 59 – Infra-estrutura para simulação do projeto</i> .....	117
<i>Figura 60 – Selecionando a máquina virtual do Ginga-NCL</i> .....	118
<i>Figura 61 – Seleção default: “Ginga-NCL Development Set-top-box”</i> .....	118
<i>Figura 62 – Tela inicial da máquina virtual Ginga-NCL</i> .....	119
<i>Figura 63 – Telas para execução do set-top-box virtual via SSH</i> .....	121
<i>Figura 64 – Console SSH</i> .....	122
<i>Figura 65 – Execução do comercial interativo do UniCEUB</i> .....	123
<i>Figura 66 – Tela da interatividade do comercial interativo do UniCEUB</i> .....	124
<i>Figura 67 – Tela de ajuda da interatividade</i> .....	125
<i>Figura 68 – Tela – Curso de Agronomia selecionado na enquete interativa</i> .....	126
<i>Figura 69 – Tela – Espaço para a digitação do email do usuário</i> .....	126
<i>Figura 70 – Tela - Exemplo de email do usuário</i> .....	127
<i>Figura 71 – Recebimento do email pela enquete interativa do UniCEUB</i> .....	127

## LISTA DE TABELAS

<i>Tabela 01 – Análise das tecnologias para o Canal de Interatividade (Fonte: Teleco, 2010).</i>	28
<i>Tabela 02 – Metas de Banda Larga – Governo Federal (Fonte: PNBL, 2010) .....</i>	30
<i>Tabela 03 – Fases distintas para a modulação 4-PSK (Fonte: Bezerra, 2008) .....</i>	52
<i>Tabela 04 – Características dos três sistemas abertos com relação à difusão terrestre (Fonte: Piccioni, 2005).....</i>	67
<i>Tabela 05 – Legenda do Mapa TV Digital no Mundo .....</i>	69
<i>Tabela 06 – Número de domicílios / aparelhos eletrônicos (Teleco, 2008) .....</i>	70
<i>Tabela 07 – Canais TV Digital em Brasília (Fonte: Própria autora).....</i>	71
<i>Tabela 08 – Equações dos sinais analógicos e dos complementos de cor (Fonte: ABNT, 2008) .....</i>	80
<i>Tabela 09 – Valores das coordenadas X e Y para as componentes vermelha, verde e azul (Fonte: ABNT, 2008).....</i>	81
<i>Tabela 10 - Resumo das características com os principais valores do SBTVD .....</i>	88
<i>Tabela 11 – Fatores propostos de modulações (Fonte: Produção Profissional, 2008).....</i>	91
<i>Tabela 12 – Tabelas PSI/SI.....</i>	94

## LISTA DE ABREVIATURAS E SIGLAS

AAC	Advanced Audio Coding
ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
ASK	Amplitude Shift Keying
ATSC	Advanced Television System Commitee
CPqD	Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações
DMB-T/H	Digital Multimedia Broadcast-Terrestrial/Handheld
DSM-CC	Digital Storage Media Command and Control
DTV	Digital Television
DVB	Digital Video Broadcasting
EBC	Empresa Brasil de Comunicação
EDTV	Enhanced Definition Television
EPG	Guia de Programação Eletrônica
FSK	Frequency Shift Keying
FPS	Frames por segundo
GIF	Graphics Interchange Format
HDTV	High Definition Television
HE-AAC	High-Efficiency Advanced Audio Coding
HTML	HyperText Markup Language
Hz / MHz	Hertz / Mega Hertz
IP	Internet Protocol
ISDB	Integrated Services Digital Broadcasting
IUT	União Internacional de Telecomunicações
ISDB-T	Integrated Services Digital Broadcasting Terrestrial
ISDB-T <sub>B</sub>	Serviço Integrado de Transmissão Digital Terrestre - Brasil
JPEG	Joint Photographic Experts Group
JVM	Java Virtual Machine
LCD	Liquid crystal display
LDTV	Low Definition Television
MAC	Multiple Sub-Nyquist Sampling Encoding
MHP	Multimedia Home Plataform
Mbps	Mega bits por Segundo
MPEG	Moving Picture Experts Group
MUSE	Multiple Sub-Nyquist Sampling Encoding

NBR	Canal de TV pertencente à EBC
NCL	Nested Context Language
NHK	Nippon Hoso Kyokai
NTSC	National Television System Committee
PAL	Phase Alternating Line
PNBL	Plano Nacional de Banda Larga
PSK	Phase Shift Keying
PUC/Rio	Pontifícia Universidade Católica do Rio de Janeiro
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
SBT	Sistema Brasileiro de Televisão
SBTVD	Sistema Brasileiro de Televisão Digital
SDTV	Standard Definition Television
SECAM	Séquentiel Couleur à Mémoire
STB	Set-top-box
TS	Transport Stream
TVDI	TV Digital Interativa
UniCEUB	Centro Universitário de Brasília
UFPB	Universidade Federal da Paraíba
UHF	Ultra High Frequency
VHF	Very High Frequency
W3C	World Wide Web Consortium
XHTML	Extensible DTV Markup Language
XHTML	Extensible Hypertext Markup Language

## RESUMO

A Televisão não poderia ficar de fora do processo de evolução tecnológica. Sendo uma das principais mídias brasileiras, a mudança para a transmissão digital trouxe mudanças em toda a sociedade. Um dos pontos principais da TV digital é a interatividade. Mais do que qualidade, quantidade ou portabilidade, é a possibilidade de interagir que modificará a maneira como o usuário assiste à televisão. Dessa forma, este projeto consiste em apresentar uma simulação de interatividade em uma transmissão de TV que utiliza um comercial do UniCEUB, sendo essa interatividade plena, que é uma interação entre o telespectador e a emissora com canal de retorno, dentro das normas da ABNT, que ditam padrões do SBTVD. Demonstrando principalmente os conceitos da linguagem NCL e o middleware Ginga, que sem sua existência, a interatividade no Sistema Brasileiro de Televisão Digital não seria possível.

Palavras-chave: TV Digital, SBTVD, ISDB-T<sub>B</sub>, Ginga, NCL, middleware, interatividade, canal de retorno, UniCEUB.

## ABSTRACT

The TV could not get out of the process of technological evolution. As a leading Brazilian media, the shift to digital broadcasting has brought changes throughout society. A major focus of digital TV is the interactivity. More than quality, quantity or portability is the ability to interact will change the way the user watches television. Thus, this project is to provide a simulation of interactivity in a TV broadcast that uses a commercial UniCEUB, and this full interactivity, which is an interaction between the viewer and the broadcast with return channel within the rules of ABNT, which SBTVD dictate standards. Showing mainly the concepts of language NCL and middleware Ginga, that without its existence, the interactivity of the Brazilian System of Digital Television would not be possible.

Keywords: Digital TV, SBTVD, ISDB-TB, Ginga, NCL, middleware, interactivity, the return channel, UniCEUB.

## **CAPÍTULO 1 - INTRODUÇÃO**

A televisão surgiu no Brasil na década de 50, com a criação da primeira emissora brasileira, a PRF-3 TV Difusora, tendo sua primeira transmissão no Palácio do Rádio, na cidade de São Paulo. (Fonte: Nova TV Digital, 2010). Com o passar dos anos, outras emissoras foram surgindo e todo um padrão de vida foi sendo transformado. Após a mudança das transmissões em preto e branco para o colorido, a sociedade passou a ser influenciada mais ativamente por esse meio de comunicação, principalmente os jovens, nas suas atitudes, valores e comportamentos. Com o avanço da tecnologia no século XX, a televisão não ficou para trás, sendo constantemente atualizada, a cada ano víamos aparelhos cada vez mais modernos. E toda essa evolução foi acompanhada pelas transmissões. Do preto e branco para o colorido, e agora do analógico para o digital. A TV Digital está trazendo novamente uma mudança de hábitos na sociedade, não apenas com melhorias significantes nas imagens, mas inclusive com possibilidades de serviços interativos, como compras, movimentações financeiras, informações mais rápidas, tudo através da televisão. Ao assistir a um programa, o telespectador terá acesso a inúmeras informações diretamente do controle remoto. Essas informações poderão tratar do programa em si, como também poderão ser pessoais, pois com o avanço da interatividade, o acesso à internet será feito pela TV, ou seja, informações pessoais, como emails e contas bancárias serão acessadas facilmente. Além dessa facilidade, haverá um contato maior entre a emissora e o telespectador, já que nessa interatividade haverá conversas e respostas do telespectador com a emissora.

E é sobre essa interatividade que trata esse trabalho. Através de um comercial do Uniceub, o telespectador responderá uma enquete e sua resposta, além de ser mostrada na tela, será também enviada por email. Assim, o trabalho aqui apresentado, tem o intuito de demonstrar que essa relação emissora e telespectador pode ser realizada de uma maneira mais rápida e intuitiva, através de programas interativos onde as respostas do telespectador serão recebidas pela emissora, pelo canal de descida, e sua confirmação de recebimento será devolvida ao telespectador pelo canal de retorno.



## 1.1. Motivação

A implantação da TV Digital no Brasil trouxe uma possibilidade de aumento da rede de distribuição da comunicação entre a população, ou seja, como a televisão está presente em 95,10% dos lares brasileiros (dados do IBGE/2008), essa tecnologia digital implica em mudança na forma de fazer e transmitir TV, variando, assim, o leque de oportunidades para as emissoras e para os telespectadores. Isso fará com que um número cada vez maior de pessoas passe a ter acessos a essa tecnologia, o que integra o objetivo da inclusão social.

Essa inclusão social será ainda mais facilitada pelo uso de softwares livre, já que o governo brasileiro é um grande incentivador dessa tecnologia. (Fonte: Portal do Software Livre, 2010). Dessa maneira, a população em geral conseguirá ter acesso a uma tecnologia mais barata, o que será de grande valia para as emissoras, pois conseguirão obter novos telespectadores e passará a existir uma concorrência de audiência mais variada e com melhor conteúdo. Com isso, o telespectador sai ganhando, pois perceberá mudanças na forma de transmissão e ainda poderá opinar com mais rapidez através do canal de interatividade.

Além de todos esses fatores acima citados, uma grande motivação é o fato de eu estar atuando diretamente na área, já que atualmente trabalho em uma emissora de televisão pública, Empresa Brasil de Comunicação, que engloba a TV Brasil (canal aberto), TV Brasil Internacional (Incorporada à TV Brasil) e a NBR (canal a cabo). A TV Brasil foi a segunda emissora de Brasília a implantar a transmissão digital. Dessa forma, participei das etapas de instalação dos transmissores digitais e estamos, a cada dia, melhorando e habilitando o serviço de EPG e interatividade com o público. Já estamos implantando (em fase de testes ainda) juntamente com a equipe da Dataprev, a TV Digital Social, pela qual o usuário poderá ter todos os serviços da Dataprev disponibilizados pela TV.

## 1.2. Objetivo Geral

Criação de uma enquete interativa em um comercial do UniCEUB no qual o telespectador receberá sua resposta via internet. Essa enquete será codificada em *Next Context Language* (NCL) que será acessada com a utilização de um set-top Box virtual (Ginga-NCL Virtual Set-top Box) que rodará em uma máquina servidor e que enviará via acesso remoto

todas as informações necessárias para a leitura e acesso dessa enquete numa segunda máquina (cliente). A resposta dessa enquete retornará a partir de um canal de retorno via internet.

### **1.3. Objetivos Específicos**

Para que este projeto obtenha êxito, foram traçados algumas metas a serem cumpridas durante a execução. São elas:

- ✓ Desenvolvimento (criação) da enquete interativa na linguagem NCL através do editor Composer;
- ✓ Instalar o set-top box virtual - Ginga-NCL Virtual Set-top Box – no servidor;
- ✓ Criar a conexão entre servidor e cliente;
- ✓ Avaliar os resultados obtidos pela utilização do Ginga;
- ✓ Avaliar a forma de resposta pelo canal de interatividade;
- ✓ Documentar todas as ações realizadas e características de respostas do processo em geral.

### **1.4. Metodologia Científica**

Esse trabalho possui sua metodologia voltada para a pesquisa teórica unida à prática de todo o seu sistema, já que não é possível a criação de um trabalho dessa magnitude com o objetivo de obter resultados práticos, sem uma pesquisa teórica.

Dessa forma, há o lado da pesquisa propriamente aplicada com o objetivo de obtenção de resultados qualitativos, ou seja, toda essa pesquisa será transformada em uma implementação prática com a demonstração de uma interatividade com o canal de retorno via internet.

### **1.5. Estrutura do Trabalho**

O projeto está dividido em seis capítulos, da seguinte forma:

- ✓ O segundo apresenta o problema, quais as etapas necessárias para se chegar ao objetivo determinado e uma breve explicação da solução em si;
- ✓ O terceiro trata do referencial teórico. Toda base teórica será listada e explicada nesse capítulo;
- ✓ O quarto demonstra a proposta de solução do problema, com todas as etapas bem definidas e explicadas;
- ✓ O quinto mostra a aplicação da solução com resultados e sua simulação; e
- ✓ O sexto traz a conclusão do trabalho com sugestão de trabalhos futuros.

### **1.6. Resultados Esperados**

Recebimento da resposta à enquete pela internet. O telespectador irá interagir com o comercial, apresentará sua resposta, que deverá retornar para o mesmo através da internet. Isso será possível graças ao desenvolvimento da implementação juntamente com a programação em NCL, para o documento hipermídia, que serão totalmente detalhadas nos capítulos a seguir.

## CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA

### 2.1. A Implantação da TV Digital no Brasil

A televisão digital tem seu início na década de 70 pelos japoneses, quando a direção da rede pública de TV do Japão Nippon Hoso Kyokai (NHK) juntamente com mais cem estações comerciais, apoiaram cientistas do NHK Science & Technical Research Laboratories que tinham o objetivo de desenvolver uma TV de alta definição (HDTV). (Fonte: Wikipédia, 2010). Na década de 80, pesquisadores da Comunidade Européia fixaram-se no desenvolvimento de um padrão único, enquanto no Japão, mais de 14 milhões de aparelhos já recebiam a HDTV na sua fase inicial. No início da nova plataforma, os avanços efetuados na Ásia e na Europa eram acompanhados de perto pelos mesmos setores nos Estados Unidos. Tanto os radiodifusores e fabricantes quanto o governo norte-americano faziam questão de encontrar uma solução nacional para a questão. Na década de 90 foram criados os padrões de compressão mais utilizados hoje em dia: MP3, MPEG-1 e MPEG-2. Esse último, tornou-se o padrão oficial dos sistemas de DVD e da TV de alta definição.

Em 1996, a TV com formato digital tem início no Brasil através de TV por assinatura via satélite como DirecTV e SKY. (Fonte: Wikipédia, 2010). Apesar da imagem ser transmitida em sinal digital, esses sistemas não permitiam a alta definição e a interatividade era bastante limitada. As discussões para a implantação da TV digital iniciaram-se em 1998. Em 1999, a Anatel, juntamente com o CPqD - Centro de Pesquisa e Desenvolvimento em Telecomunicações - deu início ao processo de avaliação técnica e econômica brasileira para a tomada de decisão quanto ao padrão de transmissão digital a ser aplicado no país. Em 2003, foi fundado o Comitê do SBTVD, responsável pelos estudos que definiriam o padrão a ser adotado. Dois anos após, o sistema foi apresentado pelo Ministério das Comunicações. Dessa forma, no ano seguinte, 2006, V. Ex<sup>a</sup>. Luiz Inácio Lula da Silva, presidente da República Federativa do Brasil, assinou o decreto 5620/2006, que trata sobre a implantação do Sistema Brasileiro de TV Digital Terrestre e seu padrão como sendo o ISDB-T<sub>B</sub>, já que foi desenvolvido com base no sistema japonês *ISDB-T - Integrated Services Digital Broadcasting Terrestrial*. Esse sistema oferece uma série de diferenciais em relação aos sistemas de TV digital atualmente em funcionamento no mundo. Esses diferenciais estão justamente na união entre a base técnica de transmissão do sistema japonês com os padrões de compressão digital

de áudio e vídeo introduzidos pelo Brasil, que são mais modernos e eficientes do que os adotados por outros padrões. Outros fatores decisivos na escolha desse padrão são: a capacidade do sistema atender a equipamentos portáteis, permitindo que o público assista à TV, por exemplo, em celulares, mini-televisores e outros dispositivos móveis, e proporcionar alta definição e interatividade para terminais fixos e móveis.

Com isso, tornou-se possível a transmissão digital em alta definição – HDTV - e em definição padrão – SDTV - transmissão digital simultânea para recepção fixa, móvel e portátil e interatividade, inclusive. Assim, no dia 02 de dezembro de 2007 iniciou-se a transmissão digital na cidade de São Paulo. Até meados de 2010, boa parte das capitais brasileiras já estão aptas para transmitirem em HDTV. Assim, o SBTVD permite uma transmissão de conteúdo de altíssima qualidade para os telespectadores e toda essa tecnologia é sem custo para a população, já que o SBTVD é um sistema de televisão digital aberto, livre e gratuito.

Abaixo, está o cronograma da implantação da TV Digital no Brasil.



Figura 01 – Cronograma TV Digital no Brasil (Fonte: Minicom, 2006)

A partir de 1º de julho de 2013, o Ministério das Comunicações somente outorgará a exploração do serviço de radiodifusão de sons e imagens para a transmissão em tecnologia digital. Isso é para que seja atingido o objetivo de encerramento das transmissões analógicas no país, a partir do dia 29 de junho de 2016.

## 2.2. O Ginga

Para que exista o desenvolvimento de aplicações interativas, é necessário que haja uma camada de software intermediário, um middleware. Essa camada é conhecida como *Ginga*. É uma especificação aberta adotado pelo Sistema Brasileiro de TV Digital e que é instalado em conversores (set-top boxes). Essa camada de software intermediária, entre o sistema operacional e as aplicações, possui duas funções principais: a primeira é tornar as aplicações independentes do sistema operacional da plataforma de hardware utilizados. A segunda é oferecer um melhor suporte ao desenvolvimento de aplicações. Ou seja, o Ginga será o responsável por dar suporte à interatividade.

Graças a vários anos de pesquisas realizadas pela Universidade Católica do Rio de Janeiro (PUC-Rio) e pela Universidade Federal da Paraíba (UFPB), o Ginga chegou a realidade constituído por um conjunto de tecnologias padronizadas e inovações brasileiras que o tornam a especificação de middleware mais avançada do mundo atualmente e a melhor solução para a realidade do país.

O sistema é subdividido em três subsistemas principais interligados, que permitem o desenvolvimento de aplicações seguindo dois paradigmas de programação diferentes. Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma será mais adequado do que o outro. (Fonte: DTV, 2010)

- ✓ Ginga-CC: (Ginga Common-Core) oferece o suporte básico para os ambientes declarativos (Ginga-NCL) e procedural (Ginga-J), de maneira que suas principais funções sejam para tratar da exibição de vários objetos de mídia, como JPEG, MPEG-4, MP3, GIF, entre outros formatos. O Ginga-CC fornece também o controle do plano gráfico para o modelo especificado para o ISDB-TB e controla o acesso ao Canal de Retorno, módulo responsável por controlar o acesso à camada de rede.

✓ Ginga-NCL: foi desenvolvido pela PUC-Rio com o objetivo de prover uma infra-estrutura de apresentação para aplicações declarativas escritas na linguagem NCL, que é uma aplicação XML com facilidades para a especificação de aspectos de interatividade, sincronismo espaço-temporal entre objetos de mídia, adaptabilidade, suporte a múltiplos dispositivos e suporte à produção ao vivo de programas interativos não-lineares. Para facilitar o desenvolvimento de aplicações Ginga-NCL, a PUC-Rio criou também a ferramenta Composer, um ambiente de autoria voltado para a criação de programas NCL para TV digital interativa. Nessa ferramenta, as abstrações são definidas em diversos tipos de visões que permitem simular um tipo específico de edição (estrutural, temporal, layout e textual). Essas visões funcionam de maneira sincronizada, a fim de oferecer um ambiente integrado de autoria.

✓ Ginga-J: foi desenvolvido pela UFPB para prover uma infra-estrutura de execução de aplicações baseadas na linguagem Java, com facilidades especificamente voltadas para o ambiente de TV digital. é dividido em três partes: as APIs (Interfaces de Programação de Aplicativos) vermelhas, inovações que dão suporte às aplicações brasileiras, em especial as de inclusão social; as APIs amarelas, também inovações brasileiras, mas que podem ser exportadas para os outros sistemas; e as APIs verdes, que seguem o núcleo comum do padrão GEM (Globally Executable MHP).

As Normas ABNT relativas ao Ginga fazem parte do grupo "*Codificação de dados e especificações de transmissão para radiodifusão digital*". Nesse grupo, as normas aprovadas e publicadas são as seguintes:

- ✓ ABNT NBR 15606-1:2007 - Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 1: Codificação de dados [2ª Ed. 13.04.2010];
- ✓ ABNT NBR 15606-2:2007 - Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 2: Ginga-NCL para receptores fixos e móveis - Linguagem de aplicação XML para codificação de aplicações [Versão corrigida: 17.04.2009];

- ✓ ABNT NBR 15606-3:2007 - Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 3: Especificação de transmissão de dados [Versão corrigida: 22.08.2008];
- ✓ ABNT NBR 15606-4:2010 - Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 4: Ginga-J – Ambiente para a execução de aplicações procedurais [1ª Ed. 13.04.2010];
- ✓ ABNT NBR 15606-5:2008 - Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 5: Ginga-NCL para receptores portáteis – Linguagem de aplicação XML para codificação de aplicações [Versão Corrigida: 14.04.2009].

Diferente dos outros sistemas, os ambientes de apresentação e execução do middleware Ginga se complementam, unidos por uma ponte em uma implementação sem nenhuma redundância, o que confere ao sistema uma ótima eficiência, tanto em termos de uso de CPU quanto de ocupação de memória.

Dessa maneira, pode-se afirmar que o Ginga é uma tecnologia que leva ao cidadão todos os meios para que ele obtenha acesso à informação, educação a distância e serviços sociais, utilizando apenas sua TV. O Ginga é uma especificação aberta, de fácil aprendizagem e livre de royalties, possibilitando que qualquer programador produza conteúdo interativo, impulsionando a programação de TVs comunitárias, por exemplo. Com o desenvolvimento do Ginga, o Brasil se tornou o primeiro país a oferecer um conjunto de soluções em software livre para TV digital.

### **2.3. O canal de Interatividade**

Canal de Interatividade é um sistema que possibilita a cada usuário, individualmente, interagir com o SBTVD, encaminhando ou recebendo informações e solicitações para os provedores de serviço e aplicações disponibilizadas pela plataforma. Segundo a recomendação J110 do ITU-T<sup>1</sup>, o Canal de Interatividade é composto por um Canal de

---

<sup>1</sup> ITU-T é a seção de Padronização da área de Telecomunicações do ITU - União Internacional de Telecomunicações ("International Telecommunication Union"), que é sediado em Genebra, Suíça.



Retorno ou caminho interativo de retorno que serve de meio de comunicação no sentido do usuário para o provedor de serviço e por um caminho interativo direto que consiste num canal individual no sentido do provedor de serviço para o usuário. É nesse Canal de Interatividade que toda e qualquer funcionalidade necessária ao estabelecimento da comunicação e transporte de informação relativa à interatividade ocorre. No caso do SBTVD, o papel de provedor de serviço será desempenhado pelas emissoras/programadoras. Portanto, o Canal de Interatividade deverá ser constituído pela interconexão das redes de televisão com as redes de telecomunicações, resultando nos dois caminhos de comunicação: o caminho interativo direto ou Canal de Descida e o caminho interativo de retorno ou Canal de Retorno.

O Canal de Descida é constituído pelos canais de radiodifusão, podendo a comunicação ser na forma Broadcast (ponto-multiponto) – aberta e disponível a todos os usuários – ou Unicast (ponto-a-ponto) – individualizada.

O Canal de Retorno é composto por qualquer combinação de tecnologias de redes de acesso de telecomunicações, desde que estabeleça a comunicação no sentido dos usuários para o provedor de serviço ou aplicação. Com a implementação do Canal de Interatividade, a comunicação poderá ocorrer no sentido emissoras/programadoras para usuário<sup>2</sup> e usuário para emissoras/programadoras, por meio da integração das redes de televisão com as redes de telecomunicações, como ilustra a figura abaixo:

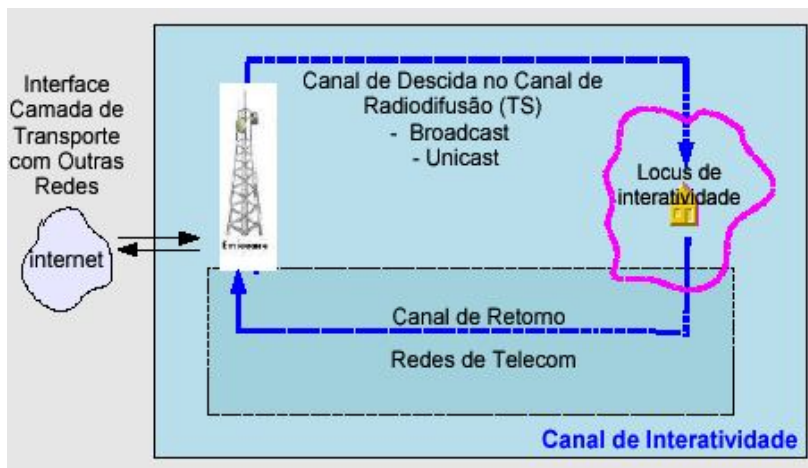


Figura 02 – Canal de Interatividade (Fonte: wirelessbrasil.org, 2010)

<sup>2</sup> No modelo convencional de televisão, a denominação aplicável seria 'telespectador'. Nesta nova concepção de televisão, o telespectador passa a ser um sujeito ativo que utiliza serviços com os quais interage, por isso é utilizado o termo 'usuário'.

## 2.4. Software Livre

Software livre, segundo a definição criada pela Free Software Foundation é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído com algumas restrições. A liberdade de tais diretrizes é central ao conceito, o qual se opõe ao conceito de software proprietário, mas não ao software que é vendido almejando lucro (software comercial). A maneira usual de distribuição de software livre é anexar a este uma licença de software livre, e tornar o código fonte do programa disponível.

Esse termo, “Software Livre”, se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro liberdades, para os usuários do software:

- ✓ A liberdade de executar o programa, para qualquer propósito (liberdade n.º. 0);
- ✓ A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade n.º. 1). Acesso ao código-fonte é um pré-requisito para esta liberdade;
- ✓ A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade n.º. 2);
- ✓ A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade n.º. 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

Há que se destacar que “Software Livre” não significa “não comercial”. Ele pode ser vendido e desenvolvido para fins comerciais, porém devido ao fato de por tantos anos o *software* ser tratado como produto é difícil não associar a denominação “livre” a de “graça”.

## 2.5. Proposta do Trabalho

A proposta deste trabalho é desenvolver uma enquete interativa para TV digital e disponibilizar sua resposta via internet. Essa enquete será um documento hipermídia cuja programação será com o uso da linguagem NCL puro, sem o uso de Java, que será disponibilizado via acesso remoto em uma máquina cliente e armazenado em um servidor. O

objetivo desse documento é permitir uma interação entre o telespectador usuário e o Uniceub, através de um comercial que será disponibilizado em uma máquina cliente. Para essa interação ocorrer, existirá um servidor que armazenará todas as informações necessárias. Dessa forma, o usuário irá responder essa enquete e receberá, via internet, a resposta dada.

## CAPÍTULO 3 - REFERENCIAL TEÓRICO

### 3.1. Interatividade

De acordo com o dicionário Aurélio, a palavra “interatividade” vem da palavra “interação”, que significa: *ação que se exerce mutuamente entre duas ou mais coisas ou pessoas; ação recíproca*. Dessa definição, vem o conceito de Interatividade na TV Digital, ou seja, a possibilidade de ter uma troca entre o telespectador e o canal de TV. Com essa tecnologia, será possível fazer compras pela TV sem ter que usar telefone, votar em pesquisas, consultar o guia de programação das emissoras, realizar operações bancárias, acessar à internet, etc.

Entretanto, esta interatividade só é possível em sua totalidade mediante a existência de um canal de interatividade. Deste modo, o estabelecimento de um meio de interação plena entre cada usuário e o sistema de televisão digital torna-se imprescindível.

Os desafios que se configuram para a implementação do canal de interatividade vão além da utilização de tecnologias pré-existentes, tal como o Sistema de Telefonia Fixa Comutada (STFC). Devem existir compromissos entre a viabilidade técnica e a econômica ao estabelecer o canal de interatividade. Os requisitos – técnicos e não técnicos – demandam identificar as diferentes tecnologias possíveis a serem empregadas, tanto as já consagradas como as que necessitariam de desenvolvimento. A diversidade e a convergência de redes e tecnologias irão requerer soluções de interconexão e de gerenciamento, além de soluções para problemas de interferências, internos a cada sistema e aqueles entre sistemas distintos. Surgirão, ainda, outras questões sérias, tais como alocação de espectro, protocolos de acesso, segurança da informação, controle de tráfego e dimensionamento de rede.

Atualmente existe uma infinidade de soluções para a implementação de um canal de interatividade e é pouco provável que algum dia seja adotada uma solução universal. Entretanto, alguns pontos devem ser levados em consideração para a escolha de uma solução para canal de interatividade, tais como: custo de utilização, custos de equipamentos para a operadora e para o usuário, existência de infra-estrutura de transmissão, mobilidade, portabilidade, área de cobertura e largura de banda, entre outros pontos que podem surgir ou serem necessários.

Tabela 01 – Análise das tecnologias para o Canal de Interatividade (Fonte: Teleco, 2010)

Tecnologia	Descrição	Vantagens	Desvantagens
Satélite	Utilização da tradicional tecnologia de satélite para canal de interatividade.	<ul style="list-style-type: none"> <li>• Cobertura de grandes áreas geográficas;</li> <li>• Concentração da totalidade dos serviços num único operador, e conseqüentemente uma maior simplicidade e uniformização dos serviços.</li> </ul>	<ul style="list-style-type: none"> <li>• Elevados custos de equipamento de transmissão;</li> <li>• Mobilidade;</li> <li>• Portabilidade;</li> <li>• Largura de banda no sentido ascendente.</li> </ul>
GSM	Utilização de GSM como canal de interatividade.	<ul style="list-style-type: none"> <li>• Tecnologia barata e disponível;</li> <li>• Portabilidade;</li> <li>• Mobilidade;</li> <li>• Elevada cobertura geográfica;</li> <li>• Uso de infra-estrutura existente;</li> <li>• Capacidade suficiente para implementar o canal de interatividade no sentido ascendente e algumas funcionalidades no sentido descendente.</li> </ul>	<ul style="list-style-type: none"> <li>• Custos mensais cobrados pelo operador de telecomunicações;</li> <li>• Possível saturação do meio disponível.</li> </ul>
Tecnologia	Descrição	Vantagens	Desvantagens
STFC Telefone	Utiliza-se de um modem sobre uma linha comum para a transmissão de dados.	<ul style="list-style-type: none"> <li>• Equipamento padrão barato e simples;</li> <li>• Baixo custo gasto por parte da operadora em equipamentos;</li> <li>• Capacidade suficiente para implementar o canal de interatividade no sentido ascendente e algumas funcionalidades no sentido descendente;</li> <li>• Existência de infra-estrutura de transmissão.</li> </ul>	<ul style="list-style-type: none"> <li>• Custos das chamadas;</li> <li>• Largura de banda suficiente para a maioria dos serviços e aplicações;</li> <li>• Ocupação da linha telefônica;</li> <li>• Mobilidade.</li> </ul>
STFC DSL	Tecnologia emergente que utiliza as antigas linhas telefônicas com uma eficiência muito superior.	<ul style="list-style-type: none"> <li>• Largura de banda elevada;</li> <li>• Não ocupa a linha telefônica podendo o usuário receber e efetuar chamadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Custo do equipamento e mensalidade;</li> <li>• Utilização restrita a centros populacionais;</li> <li>• Mobilidade;</li> <li>• Portabilidade.</li> </ul>
Cabo	Tecnologia que utiliza de cabos coaxiais ou de fibras ópticas para transmissão de dados.	<ul style="list-style-type: none"> <li>• Largura de banda elevada;</li> <li>• Concentração da totalidade dos serviços num único operador, e conseqüentemente uma maior simplicidade e uniformização dos serviços.</li> </ul>	<ul style="list-style-type: none"> <li>• Usuário precisa de acesso à operadora de cabo;</li> <li>• Custo de mensalidades;</li> <li>• Mobilidade;</li> <li>• Portabilidade.</li> </ul>
LMDS	Sistema baseado em ligações de microondas de curto alcance nas bandas 950 a 2150 MHz.	<ul style="list-style-type: none"> <li>• Largura de banda elevada;</li> <li>• Não requer a instalação de cabos até a casa do usuário.</li> </ul>	<ul style="list-style-type: none"> <li>• Necessidade de construir uma infra-estrutura nova para suportar esta tecnologia;</li> <li>• Funcionamento restrito à linha de visada.</li> </ul>
DECT	DECT é uma especificação para uma rede de acesso sem fios que trabalha nas frequências entre 1880 – 1900 MHz.	<ul style="list-style-type: none"> <li>• Capacidade suficiente para implementar o canal de interatividade no sentido ascendente e algumas funcionalidades no sentido descendente;</li> <li>• Mobilidade.</li> </ul>	<ul style="list-style-type: none"> <li>• Não é indicado para longas distâncias;</li> <li>• Requer a construção de uma infra-estrutura;</li> <li>• Funcionamento restrito à linha de visada.</li> </ul>

O caso brasileiro é atípico, pois o SBTVD propõe uma concepção diferente para o sistema de TV digital, atribuindo-lhe uma forte função social como agente de inclusões digital e social, tendo um importante caráter educativo, de saúde e de prestação de serviços públicos, além da sua função de entretenimento. Com a exigência dessas características, o governo federal pretende criar possibilidades para que toda a população tenha acesso a essas novas tecnologias. Entretanto, para isso ocorrer, o Ministério das Comunicações estabeleceu, nesse ano, uma proposta para um Plano Nacional de Banda Larga (PNBL), com o objetivo de massificar, até 2014, a oferta de acessos banda larga e promover o crescimento da capacidade da infra-estrutura de telecomunicações do país.

Esse Plano Nacional de Banda Larga surgiu em um momento extremamente favorável para a questão da TVDI – TV Digital Interativa, já que a melhor tecnologia a ser empregada no uso do canal de retorno, sendo analisado inclusive custo X benefício, é justamente a banda larga.

De acordo com o documento oficial do PNBL, o número de acessos a Internet em banda larga fixa atingiu aproximadamente 9,6 milhões em dezembro de 2008, o que corresponde a aproximadamente 17,8 acessos a cada 100 domicílios e 5,2 acessos a cada 100 brasileiros. Desta maneira, este PNBL deverá elevar a quantidade de acessos banda larga para cerca de 30 milhões fixos e de aproximadamente 60 milhões de acessos móveis, totalizando 90 milhões de acessos banda larga. Isto representa alcançar um nível de tele densidade próximo de 50 acessos por 100 domicílios (em acessos fixos banda larga), ou 45 acessos por 100 habitantes (acessos fixos e móveis em banda larga) no total. Abaixo, vemos a tabela de metas de Banda Larga do governo federal.

Tabela 02 – Metas de Banda Larga – Governo Federal (Fonte: PNBL, 2010)

Abrangência e tipo de acesso	Metas para 2014
Acesso Fixo Individual (Urbano e Rural)	<ul style="list-style-type: none"> <li>• <b>30 milhões de acessos banda larga fixa</b> (urbanos e rurais), somando-se os acessos em domicílios, propriedades, empresas e cooperativas.</li> </ul>
Acesso Fixo Coletivo (Urbano e Rural)	<p>Levar acesso banda larga a 100% dos órgãos de Governo, incluindo:</p> <ul style="list-style-type: none"> <li>• 100% das unidades da Administração Federal, dos Estados e Municípios.</li> <li>• 100% das escolas públicas ainda não atendidas (mais de 70.000 rurais).</li> <li>• 100% das unidades de saúde (mais de 177.000).</li> <li>• 100% das bibliotecas públicas (mais de 10.000).</li> <li>• 100% dos órgãos de segurança pública (mais de 14.000).</li> </ul> <p>Implantar 100 mil novos Telecentros Federais até 2014.</p>
Acesso Móvel	<ul style="list-style-type: none"> <li>• <b>60 milhões de acessos banda larga móvel</b>, entre terminais de voz/dados (com serviço de dados ativo) e modems exclusivamente de dados.</li> </ul>

Assim, o Canal de Interatividade no SBTVD não deverá ser visto como um simples canal de retorno, tal qual ocorre em outros países, mas principalmente como um meio de comunicação para a promoção das inclusões digital e social, onde as pessoas possam efetivamente interagir com o sistema de TV, usufruindo dos serviços de utilidade pública, sociais e educativos acima citados, criando uma ponte entre o “mundo broadcast” e o “mundo telecom”.

### 3.2. Bidirecionalidade

O canal de interatividade só é possível se houver uma comunicação plena entre o telespectador e a emissora. E isso é basicamente o início do conceito de comunicação bidirecional.

A conjunção de várias tecnologias, tais como televisão, analógica e digital, com a computação, interligada à internet, proporciona essa bidirecionalidade. Ela passa a ser uma variável para que essa tecnologia da TV Digital cumpra o seu objetivo. Conforme explica Silva (SILVA, Marco, 1995: 07-08):

*“Só existe comunicação a partir do momento em que não há mais nem emissor nem receptor e, a partir do momento que todo emissor é potencialmente um receptor e todo receptor é potencialmente um emissor. Portanto, comunicação é bidirecionalidade entre os pólos emissor e receptor, ou seja, comunicação é troca entre codificador e decodificador sendo que cada um codifica e decodifica ao mesmo tempo.”*

### **3.3. A Linguagem NCL**

A linguagem NCL - *Nested Context Language* - é uma linguagem declarativa para autoria de documentos hipermídia baseados no modelo conceitual NCM - *Nested Context Model* - e foi desenvolvida utilizando uma estrutura modular, seguindo os princípios adotados pelo W3C<sup>3</sup>. Sua estrutura é baseada em XML, dessa forma, seus módulos para especificação de conectores e templates de composição, chamados XConnector e XTemplate respectivamente, podem ser incorporados a outras linguagens existentes, como por exemplo, XLink, XHTML e SMIL, usadas para autoria de documentos na Web.

Por essas características, a União Internacional para as Telecomunicações (UIT) - agência regulatória pertencente às Nações Unidas para questões de telecomunicações e tecnologia da informação - certificou em 29 de abril de 2009 o ambiente Ginga-NCL e a linguagem de script Lua como a primeira recomendação internacional para suporte à interação e multimídia para dispositivos de IPTV, como "Recomendação H.761".

✓ Estrutura básica de uma arquivo ncl:

A exemplo da linguagem html, a linguagem ncl caracteriza-se por ser uma linguagem de marcação, possuindo elementos (tags) padrões assim como o html.

A seguir, vamos demonstrar, através de alguns exemplos simples, a estrutura básica existente em um código de arquivo ncl:

---

<sup>3</sup> W3C: Consórcio World Wide Web. É uma comunidade internacional que desenvolve padrões com o objetivo de garantir o crescimento da web.



→ Estrutura inicial do arquivo ncl:

O cabeçalho de um arquivo ncl possui dois elementos básicos:

**1) <?xml version="1.0" encoding="ISO-8859-1">** - Essa é normalmente a primeira linha de um código ncl, onde são definidas a versão do xml bem como a codificação. Isto é, define o tipo de caracteres que serão exibidos na tela. No nosso caso, obviamente, devemos usar a codificação "ISO-8859-1", que corresponde aos caracteres ocidentais.

**2)<ncl id="new\_ncl\_file" xmlns= "http://www.ncl.org.br/**

**NCL3.0/EDTVProfile">** - Essa tag, a exemplo da tag <html>, define o início do programa ncl propriamente dito, que é conhecido como *nó ncl*, simbolizando toda a aplicação NCL. Ele possui dois atributos: 'id' e 'xmlns'. O 'id' é utilizado para identificar a sua aplicação, já o 'xmlns' serve para identificar o perfil NCL que está sendo usado. Dentro do NCL temos dois nós obrigatórios: 'head' e 'body' que serão analisados abaixo:

→ Cabeçalho do programa:

Abaixo da estrutura inicial do arquivo ncl, encontra-se o cabeçalho do programa. Assim como no html, esse cabeçalho é delimitado pelas tags <head></head>. Dentro do cabeçalho do programa (entre as tags <head></head>), encontram-se três elementos básicos de um programa ncl. São os seguintes elementos:

**1) Base de regiões:** As regiões são tags que ficam dentro do cabeçalho("head"). Elas servem para especificar o posicionamento das mídias (vídeo, texto, imagens) na tela. Outra característica que as regiões têm é de indicar em qual dispositivo aquela aplicação que você está fazendo será exibida, pois no Ginga será possível interagir com sua TV através de outros dispositivos como o celular.

Algumas características das regiões é que suas dimensões podem ser declaradas com valores absolutos (ex: 50 pixels) ou valores relativos ao tamanho da tela (em porcentagem). Uma característica muito importante das regiões é que elas podem ser aninhadas, o que significa que você pode declarar regiões dentro de outras regiões, onde as regiões declaradas dentro de outras tem como limite as dimensões de sua região pai (se declarei uma região 1 com 20% de largura e altura da tela total, ao declarar uma região 2 dentro da 1, as dimensões dela serão baseadas nas dimensões da região 1).

A estrutura para criarmos uma região é a seguinte:

```

1 <head>
2   <regionBase>
3     <region>
4       <region>
5         </region>
6       </region>
7     </regionBase>
8 </head>

```

Figura 03 – Estrutura de Regiões em um arquivo NCL

Dentro da “head” é aberta uma tag “regionBase” (base de regiões), pois podemos criar  $n$  regiões. Essa tag não necessita de nenhum atributo, sendo necessário apenas se for utilizar multiDispositivos, o que não será abordado aqui neste trabalho.

Dentro da base de regiões, podemos adicionar nossa região com a tag “region”. Como já foi dito, regiões podem ser aninhadas, podendo assim adicionar regiões dentro de outras.

A região mais externa (cujo pai é a “regionBase”) tem suas dimensões limites a resolução da tela do dispositivo que está sendo usado, enquanto as outras regiões que foram aninhadas tem como limite a região pai logo acima dele.

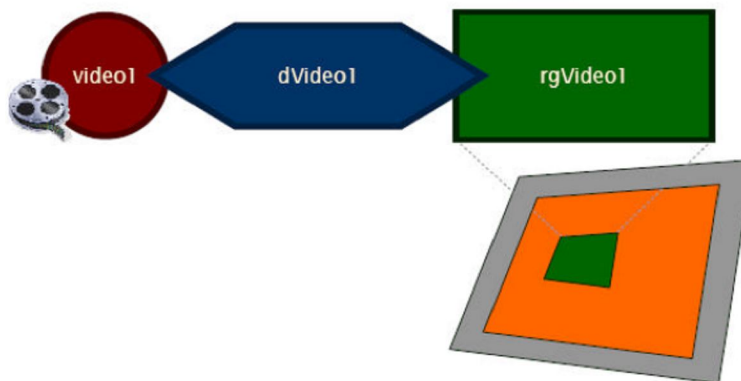


Figura 04 – Região do Programa NCL

→ Os Atributos das Regiões:

- Id: identifica a região de forma única em todo programa. Ele deve ser único em todo programa, nenhum outro nó pode ter um id igual;
- Left: Posiciona a margem esquerda da região na distância especificada a partir da esquerda da região-pai. Pode ter valores percentuais não-negativos, ou unidades de pixels;

- **Right:** Posiciona a margem direita da região na distância especificada a partir da direita da região-pai. Pode ter valores percentuais não-negativos, ou unidades de pixels;
- **Top:** Posiciona a margem superior da região na distância especificada a partir da margem superior da região-pai. Pode ter valores percentuais não-negativos, ou unidades de pixels;
- **Bottom:** Posiciona a margem inferior da região na distância especificada a partir da margem inferior da região-pai. Pode ter valores percentuais não-negativos, ou unidades de pixels;
- **Height:** Define a altura de uma região. Pode ter valores percentuais não-negativos, ou unidades de pixels;
- **Width:** Define a largura de uma região. Pode ter valores percentuais não-negativos, ou unidades de pixels;
- **zIndex:** Especifica a precedência de sobreposição da região. Regiões com maiores valores de zIndex devem ser obrigatoriamente empilhadas no topo de regiões com valores de zIndex menores. Se duas regiões tiverem a mesmo valor de zIndex vale a ordem temporal.

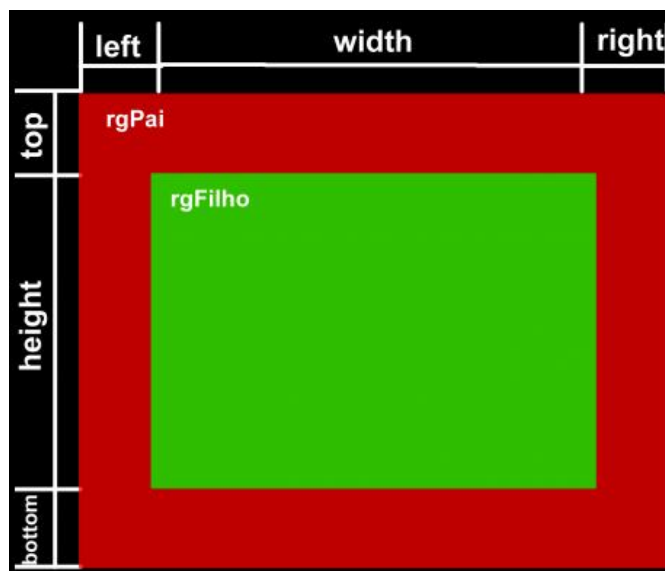


Figura 05 – Atributos das regiões em um arquivo NCL (Fonte: dtvgames, wordpress, 2010)

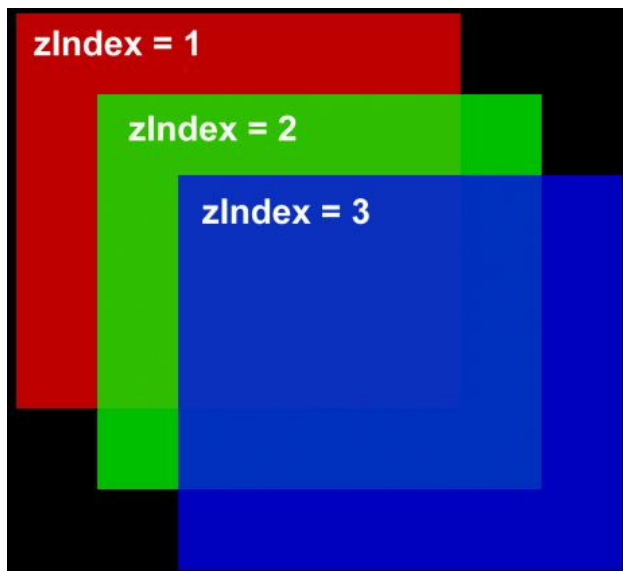


Figura 06 – Exemplificação do atributo zIndex NCL (Fonte: dtvgames, wordpress, 2010)

**2) Base de descritores:** Os descritores definem como as mídias serão apresentadas, isto significa que quando formos declarar uma mídia teremos que relacioná-la a um descritor, e este descritor irá apresentar algumas características que auxiliarão na exibição da mídia. Os descritores, por exemplo, determinam em qual região a mídia será exibida, o que acontece ao final da apresentação dela, efeitos de transição e também propriedades como volume, transparência e outros.

Outra característica importante dos descritores, é que com eles podemos fazer a navegação de nossa aplicação. No descritor podemos declarar um identificador do foco que será utilizado para identificar qual mídia está em foco no momento. O foco pode ser usado para a navegação em uma aplicação, mas também pode ser útil para dizer a quem pertence o controle do controle remoto, que no caso de usarmos aplicações Lua é importante, pois por esta aplicação podemos querer controlar a interação com o usuário, dentre outras possibilidades. Uma importante observação: os descritores não podem ser aninhados.

A estrutura para declararmos os descritores é a seguinte:

```

1 <descriptorBase>
2   <descriptor id="" >
3     <descriptorParam name="" value=""/>
4   </descriptor>
5 </descriptorBase>

```

Figura 07 – Estrutura dos descritores em um arquivo NCL (Fonte: adaptado,dtvgames, wordpress, 2010)

→ Os Atributos dos Descritores:

- Id: Identifica univocamente, ou seja, admite-se uma única forma de interpretação, o elemento dentro do documento NCL. Seu valor é um identificador XML;
- explicitDur: Diz quanto tempo a exibição da mídia deve durar;
- region: Refere-se à região definida pelos elementos <region> do <regionBase>;
- freeze: Especifica o que acontece com um objeto ao final de sua apresentação;
- transIn: Referencia uma transição que deverá iniciar no começo da duração ativa dos elementos de mídia;
- transOut: Referencia uma transição que deverá iniciar no final da duração ativa dos elementos de mídia.

Apenas o id e a região tem obrigatoriedade, todas os demais atributos são opcionais de acordo com a aplicação. Os seguintes atributos são utilizados para manipulação do foco (seu uso depende do tipo da aplicação):

- moveLeft: Especifica para qual mídia irá o foco ao se apertar a seta esquerda quando este descritor estiver ativo;
- moveRight: Especifica para qual mídia irá o foco ao se apertar a seta direita quando este descritor estiver ativo;
- moveDown: Especifica para qual mídia irá o foco ao se apertar a seta para baixo quando este descritor estiver ativo;
- moveUp: Especifica para qual mídia irá o foco ao se apertar a seta para cima quando este descritor estiver ativo;
- focusIndex: Especifica um índice sobre o qual o foco pode ser aplicado, quando esse elemento estiver em exibição, utilizando o elemento <descriptor> que definiu o atributo;
- focusBorderColor: Define a cor de destaque(quando o elemento estiver em foco) e pode receber os nomes reservados de cor: “white”, “black”, “silver”, “gray”, “red”, “maroon”, “fuchsia”, “purple”, “lime”, “green”, “yellow”, “olive”, “blue”, “navy”, “aqua”, ou “teal”;

- `focusBorderWidth`: Define a largura de destaque (quando o elemento estiver em foco);
- `focusBorderTransparency`: Define a transparência de destaque (quando o elemento estiver em foco);
- `focusSrc`: Especifica um conteúdo alternativo a ser apresentado, ao invés do conteúdo da apresentação atual, se um elemento receber o foco;
- `focusSelSrc`: Especifica um conteúdo de mídia alternativo a ser apresentado, quando um elemento em foco é selecionado pressionando a tecla de ativação;
- `selBorderColor`: Especifica uma cor de borda, quando selecionada;

Como foi visto acima, os descritores podem possuir parâmetros. Estes parâmetros (`descriptorParam`) possuem dois atributos:

- `Name`: Indica o nome do parâmetro/atributo;
- `Value`: Especifica um valor inicial para o parâmetro/atributo definido;

Os valores variam dependendo do tipo da mídia que você irá usar. Alguns exemplos são:

✓ Para Som:

→ `“soundLevel”, “balanceLevel”, “trebleLevel”, “bassLevel”`.

✓ Para Mídias Visuais:

→ `“top”, ”left”, ”bottom”, ”right”, ”width”, ”height”, “zIndex”, “location”, “size”(w,h), “bounds”(l,t,w,h), “backGround”, “visible”, “transparency”, “fit”(fill, hidden, meet, meetBest, slice) .`

**3) Base de conectores:** No cabeçalho do programa, a exemplo da base de regiões e da base de descritores, também é definida a Base de conectores. Seguindo a mesma lógica das bases anteriores, a base de conectores começa com a tag `<connectorBase>` e termina com a tag `</connectorBase>`.

Os conectores definem como os elos são ativados e o que eles disparam. Por exemplo: Através de conectores, é possível fazer com que a exibição de uma mídia comece simultaneamente a outra, bem como termine simultaneamente a outra e diversas outras ações.

Normalmente os conectores estão definidos em um arquivo exterior ao código, semelhante a um arquivo .css. Há a possibilidade de criar os próprios conectores na base de conectores, entretanto é recomendado o uso do arquivo externo já que ele contém dezenas de conectores prontos para o uso diminuindo-se, assim, o trabalho.

- Exemplo de carregamento de uma base de conectores externa:

```
<connectorBase>
  <importBase alias="connectors"
                                documentURI="connectorBase.ncl"/>
</connectorBase>
```

O arquivo "connectorBase.ncl", que está no mesmo diretório do programa ncl e contém dezenas de conectores prontos, é carregado no programa através do atributo "documentURI". O atributo "alias" funciona como uma espécie de identificador para a base carregada. Essa "identificação" será utilizada pelos elos para fazer referência a base carregada.

→ Corpo do programa: No corpo do programa são inseridos os nós (mídias), portas, elos, âncoras, etc.

→ **NÓS:**

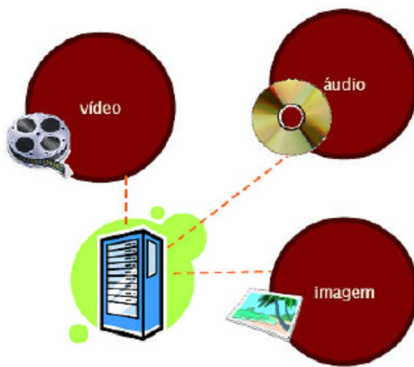


Figura 08 – Nós do programa NCL

Os nós são referências para as mídias. É todo o conteúdo no programa interativo. O nó funciona semelhante à tag <a> do html. Sendo assim, os nós funcionam como links que, quando ativados, disparam a exibição de uma determinada mídia.

Os nós podem ser de dois tipos: Nó de conteúdo (ou de mídia) e nó de composição (ou contexto).

1) Nó de conteúdo: Esse tipo de nó é associado a um tipo de mídia (txt, html, jpeg, mpeg etc). Um nó de conteúdo é iniciado através da tag <media>.

- Exemplo de utilização de um nó de conteúdo:

```
<media type="video/mpeg" id="video1" src="video1.mpg" descriptor="dVideo1">
```

Neste exemplo, é criado um nó de conteúdo fazendo referência, através do atributo "src", a um arquivo chamado video1.mpg que se encontra no mesmo diretório do programa ncl.

Na criação de um nó também deve ser determinado qual descritor regerá o seu funcionamento (onde e como será exibido). O descritor poderá, por exemplo, determinar a região da tela na qual a mídia será exibida.

Também é definido a atributo "type", que deve receber o tipo da mídia que será executada. Os tipos mais comuns são:

- image/gif;
- image/jpeg;
- image/bmp;
- video/mpeg;
- text/plain;
- text/html;
- audio/mp3.



## 2) Nó de contexto:

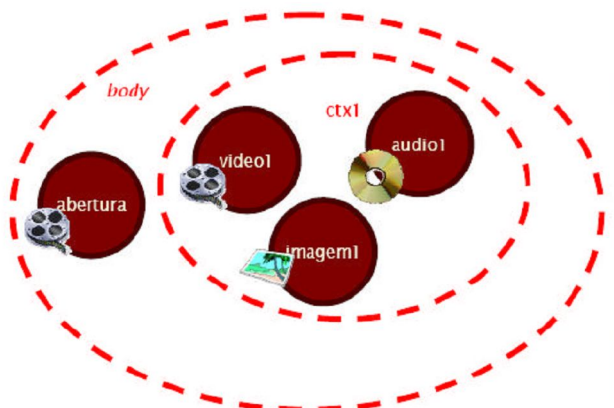


Figura 09 – Nó de Contexto do programa NCL

Os nós de contexto são compostos pelos componentes abaixo:

**- PORTAS:**

A porta, representada pela tag <port>, é um ponto de interface de um contexto. Através dela podemos ter acesso ao conteúdo de um contexto. Sendo assim, para um elo apontar para um nó interno ao contexto, este contexto deve possuir uma porta que leve ao nó interno.

- Exemplo do uso de uma porta:

```
<port id="pVideo1" component="video1" />
```

Acima, o atributo "id" define o nome da porta e o atributo "component" define que nó será acessado em um determinado contexto. Nesse caso seria ativado um nó com id igual a video1. A figura 10 exemplifica.

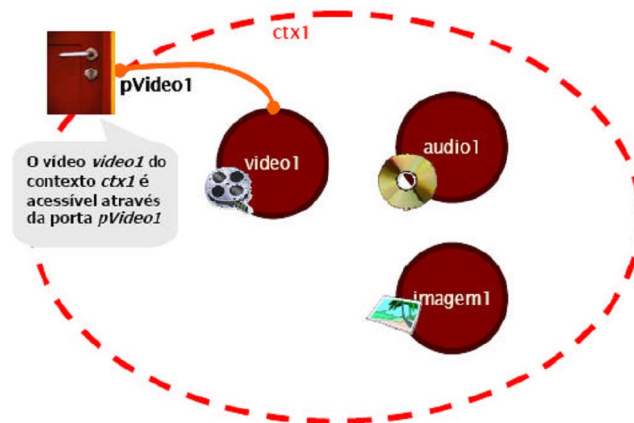


Figura 10 – Porta do programa NCL

### - ELOS:

Os elos funcionam semelhantes a um link em um arquivo html. Através do elo os possíveis eventos são sincronizados em um programa ncl. Por exemplo: iniciar a execução de uma mídia simultaneamente a outra, finalizar uma mídia imediatamente após o término de outra, etc.

Tal sincronia se torna possível através dos conectores que podem ser criados dentro do arquivo ncl ou importados de um arquivo externo.

Um elo inicia com a tag <link> e termina com a tag </link>.

- Exemplo do uso de elo:

```
<link xconnector="connectors#onBegin1StartN" id="Titulo1Inicia">
  <bind role="onBegin" component="suporte"/>
  <bind role="start" component="titulo"/>
</link>
```

O atributo "xconnector" faz referência à base de conectores que será utilizada. Observe que o atributo "xconnector" recebe dois valores separados pelo caracter "#". O primeiro valor "connectors" faz referência a base de conectores onde está o conector que será utilizado (atributo "alias" na base de conectores). A segunda parte (onBegin1StartN) indica que conector contido no arquivo importado será usado.

A tag <bind>, determina os papéis de cada nó nesse elo. No primeiro bind é determinado de que forma e qual nó ativará o elo. Nesse bind, o nó "suporte", quando executado (onBegin), fará com que, no segundo bind, o nó de mídia "título" também seja iniciado (start). Dessa forma, as duas mídias seriam exibidas simultaneamente.

### - ÂNCORAS:

As âncoras são pontos de entrada para os nós de mídia ou contextos. O objetivo de se utilizar âncoras é utilizar segmentos de um nó de mídia ou contexto, seja como origem ou destino de elos.

Existem dois tipos de âncoras: âncora de conteúdo (content anchor) e âncora de atributo (attribute anchor).

**1) Âncora de conteúdo:** define um segmento da mídia (intervalo de tempo e/ou região no espaço) que poderá ser utilizado como ponto de ativação de elos. Cada nó de mídia é composto de unidades de informação (information units). A definição dessas unidades de informação depende do tipo de mídia representado pelo nó. As unidades de informação de um vídeo, por exemplo, podem ser os frames do vídeo. Uma âncora consiste numa seleção contígua de unidades de informação de um nó.

Uma âncora de conteúdo é definida como uma tag <area> dentro da tag <media>.

Lembrando que para fazer um comentário no código NCL, deve se usar <!-- para abrir e iniciar o comentário e, para fechar e finalizar usa-se -->.

- Exemplo de utilização de âncora de conteúdo:

```
<media type="vídeo/mpeg" id="video1" src="video1.mpg" descriptor="dVideo1">
    <area id="aVideoLegenda01" begin="5s" end="9s"/>
    <area id="aVideoLegenda02" begin="10s" end="14s"/>
    <area id="aVideoLegenda03" begin="15s" end="19s"/>
</media>
```

São definidas três âncoras de conteúdo para um nó de vídeo, que são utilizadas para sincronizar legendas com o vídeo. O atributo “begin” define o tempo transcorrido na execução do vídeo (Xs) no qual a âncora será ativada. O atributo “end” funciona semelhante ao begin, determinando o tempo em que a âncora será encerrada.

2) Uma âncora de atributo: refere-se a atributos de um nó de origem ou de destino, que podem ser manipulados pelos elos. Exemplos de atributos:

- Volume de áudio de um nó de áudio ou vídeo;
- Coordenadas e dimensões de exibição de um nó de mídia visual.

Uma âncora de atributo é definida através da tag <attribute> dentro da tag <media> ou <context>. A seguir são definidas quatro âncoras de atributo para um nó de vídeo, além de uma âncora de conteúdo:

```
<media type="video" id="video1" src="media/video1.mpg" descriptor="dVideo1">
<!-- âncoras de atributos que serão manipulados pelos elos -->
  <attribute id="top" name="top"/>
  <attribute id="left" name="left"/>
  <attribute id="width" name="width"/>
  <attribute id="height" name="height"/>
<!-- âncora de conteúdo no vídeo que deve sincronizar a imagem -->
  <area id="aVideo1Imagem1" begin="3s" end="8s"/>
</media>
```

Abaixo, na figura 11, veremos uma estrutura básica de um arquivo NCL mais detalhada:

cabeçalho do arquivo NCL	1: <?xml version="1.0" encoding="ISO-8859-1"?> 2: <ncl id="exemplo01" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd">	<b>1</b>
cabeçalho do programa	3: <head>	
base de regiões	4: <regionBase> 5: <!-- regiões da tela onde as mídias são apresentadas --> 6: </regionBase>	<b>2</b>
base de descritores	7: <descriptorBase> 8: <!-- descritores que definem como as mídias são apresentadas --> 9: </descriptorBase>	<b>3</b>
base de conectores	10: <connectorBase> 11: <!-- conectores que definem como os elos são ativados e o que eles disparam --> 12: </connectorBase>	<b>8</b>
	13: </head>	
corpo do programa	14: <body>	
ponto de entrada no programa	15: <port id="plnicio" component="ncPrincipal" interface="Inicio"/>	<b>5</b>
conteúdo do programa	16: <!-- contextos, nós de mídia e suas âncoras, elos e outros elementos -->	<b>4</b>
	17: </body>	<b>6</b> <b>7</b>
término	18: </ncl>	

- 1** os cabeçalhos básicos do arquivo NCL e do programa;
- 2** as regiões da tela onde aparecerão os elementos visuais (**regionBase**);
- 3** como e onde os nós de mídia serão exibidos, através de descritores (**descriptorBase**);
- 4** o conteúdo (nós de mídia - **media**) e a estrutura (contextos - **context**) do documento (seção **body**), associando-os aos descritores;
- 5** a porta de entrada do programa, apontando para o primeiro nó a ser exibido, bem como as portas para os contextos, visando à construção dos elos entre contextos e nós de mídia (**port**);
- 6** âncoras para os nós de mídia, visando à construção dos elos entre nós (**area** e **attribute**);
- 7** elos para o sincronismo e interatividade entre os nós de mídia e contextos (**link**); e
- 8** os conectores que especificam o comportamento dos elos do documento (**connectorBase**).

Figura 11 – Estrutura Básica de um arquivo NCL (Fonte: Tutorial NCL, 2007)

### 3.4. A Linguagem Lua

Lua é uma linguagem de programação rápida e leve, projetada para estender aplicações. Combina sintaxe simples para programação procedural com poderosas construções para descrição de dados baseadas em tabelas associativas e semântica extensível. Tem sua ênfase em jogos e sistemas embutidos, como por exemplo, o middleware Ginga.

Lua é inteiramente projetada, implementada e desenvolvida no Brasil, por uma equipe da PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro). Lua nasceu e cresceu no Tecgraf, o Grupo de Tecnologia em Computação Gráfica da PUC-Rio. Atualmente, Lua é desenvolvida no laboratório Lablua. Tanto o Tecgraf quanto Lablua são laboratórios do Departamento de Informática da PUC-Rio.

Será utilizada no projeto para a obtenção de digitação do email, via chamadas no próprio arquivo NCL e para a realização da conexão TCP e chamada ao PHP, para obtermos assim, o envio do email, objetivando a criação do canal de retorno.

### 3.5 Hipertexto

Através da etimologia, ou seja, o estudo da origem das palavras, tem-se que o termo *hipertexto* é composto inicialmente pelo prefixo *hiper*, que vem do grego e significa *sobre, além*, remetendo à superação das limitações da linearidade, não havendo uma sequência em relação ao antigo texto escrito.

O termo hipertexto foi criado na década de 60 por Theodor Holm Nelson, para exprimir a idéia de escrita e leitura não linear, em um sistema de informática. Tecnicamente, um hipertexto é um conjunto de dados ligados entre si por conexão (links), que podem ser palavras, imagens, gráficos, sequências sonoras, etc. Na definição de Nelson, há a possibilidade de um nó se bifurcar e apresentar mais de uma opção ao leitor. No hipertexto não há uma única maneira de ler conteúdo, nele há múltiplas entradas e múltiplas formas de prosseguir.

Dee-Lucas (Dee-Lucas, 1996) estudou três formas de apresentação de hipertextos:

- ✓ Distribuição hierárquica;
- ✓ Lista;
- ✓ Texto convencional.

A figura 12 mostra os três tipos de apresentação de hipertextos. Cada apresentação tem uma aplicação específica, sendo possível apresentar hipertextos das três formas.

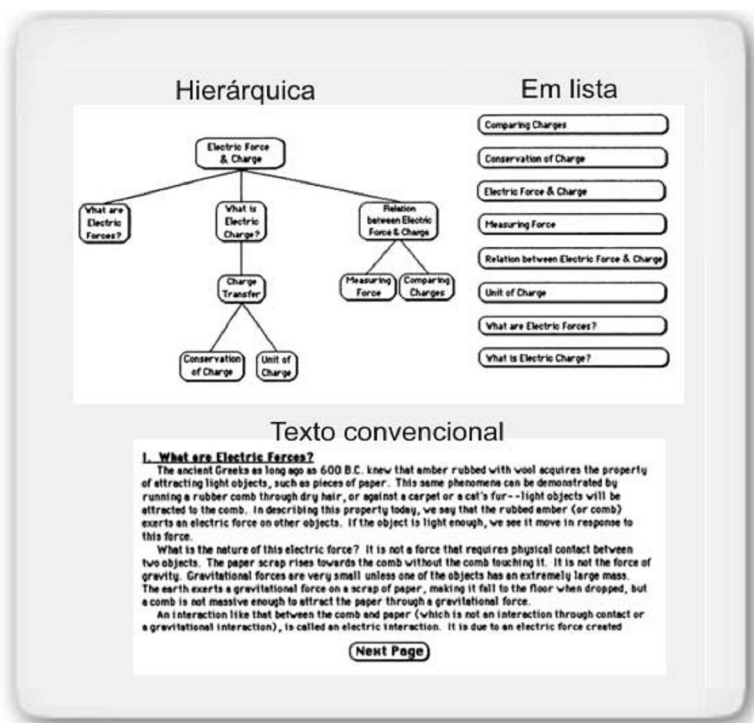


Figura 12 – Formas de apresentação de hipertextos (Fonte: DEE-LUCAS, 1996).

### 3.6. Hipermídia

Hipermídia é a reunião de várias mídias num suporte computacional, suportado por sistemas eletrônicos de comunicação. (Laufer & Scavetta, 1997).

Uma das características de não-linearidade, hipertexto, interface e multimídia numa só linguagem. Segundo Bugay (2000), uma forma bastante comum de Hipermídia é o hipertexto, no qual a informação é apresentada ao usuário sob a forma de texto, através de uma tela do

computador. O usuário pode iniciar uma leitura de forma não linear, ou seja, escolhe entre o início, meio ou fim de um texto. Segundo o autor citado, a Hiperímia pode ser considerada uma extensão do hipertexto, entretanto, inclui além de textos comuns, desde sons, animações e vídeos, e de uma forma interativa, com apenas um clicar de botão, o computador responde ao caminho desejado.

Para uma melhor diferenciação, cita-se a comparação entre a programação televisiva disponibilizada aos telespectadores hoje e a internet com as hiperímias. Na televisão, o usuário assiste à programação com horários pré-definidos, seguindo uma sequência lógica das informações que são apresentadas, tendo que assistir ao conteúdo da maneira integral, sem possibilidades de mudanças. Na internet, isso não ocorre, já que o mesmo usuário pode ler notícias na ordem e horários que ele desejar, inclusive possui a possibilidade de compartilhar aquela notícia passando o link da notícia, ou seja, o endereço do nó que se encontra aquela informação.

### **3.7. Transmissão Analógica X Digital**

A Transmissão de dados é comumente classificada em dois modos: Analógico e Digital. Neste item, serão apresentados conceitos básicos de técnicas de transmissão e multiplexação.

#### **3.7.1 Digitalização**

A digitalização consiste na conversão de um sinal analógico em um sinal composto por dois níveis, popularmente conhecidos como “zeros” e “uns”. A informação passa a estar contida na amplitude relativa do sinal e não mais na sua forma de onda. A conversão analógico/digital normalmente se faz em três etapas:

- ✓ Amostragem ou Sampling: É a representação do sinal em intervalos discretos no tempo;
- ✓ Quantização: Onde são atribuídos valores para cada amplitude do sinal amostrado;



- ✓ Codificação: É a conversão para o número binário. Para a conversão, o método de *oito bits* é o mais usado, podendo fornecer 256 níveis de sinal.

Abaixo, vemos o processo de digitalização:

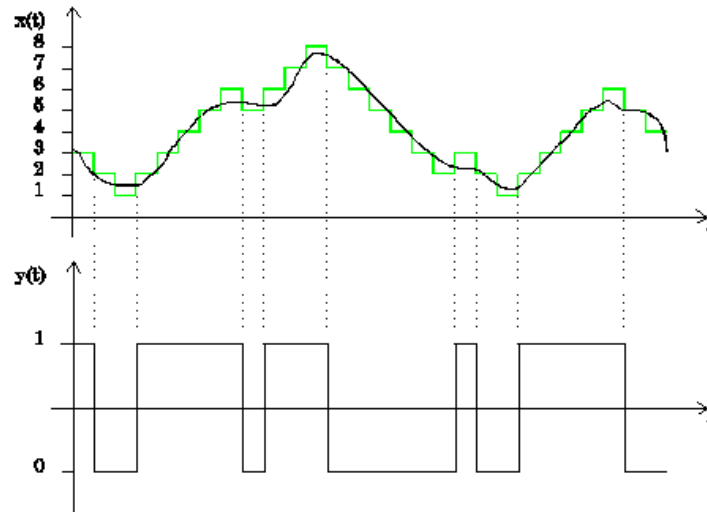


Figura 13 – Processo de Digitalização

### 3.7.2 Analógico X Digital

A transmissão das imagens e sons da televisão é feita por ondas eletromagnéticas, cuja frequência é medida em *Hertz*. A largura da banda (faixa) de transmissão é de 4 Mhz (megahertz), ou seja, 4 milhões de oscilações por segundo. Com a somatória da banda de vídeo com a de áudio e sincronismo - *sync* - temos um total de 6 Mhz.

A faixa eletromagnética da transmissão vai de 52 Mhz até 890 Mhz, sendo de 52 a 216 Mhz para as emissoras de *VHF* destinados aos canais de 2 a 13, sofrendo entre os canais 5 e 6 um intervalo para as frequências de 88 a 108 Mhz para *FM*. As emissoras de *UHF* usam frequências de 216 a 800 Hz.

Caminhando praticamente em linha reta, as ondas de televisão, sofrem reflexões ao rebater em prédios, daí os famosos fantasmas e chuviscos na imagem. A transmissão das ondas chega a um raio de 100 km, contando a partir do ponto central do transmissor.

Como a transmissão do sinal digital pelo SBTVD é feito em UHF, a faixa de frequência utilizada vai de 300Mhz até 3GHz. Contudo, como estamos tratando agora com dados (bits), usa-se mais a designação de transmissão de dados, que é feito por bps (bits por segundo). Assim, a banda foi dividida da seguinte forma: 15Mbps para vídeo, 2Mbps para áudio e 2Mbps para transmissão de dados, somando assim, 19Mbps.

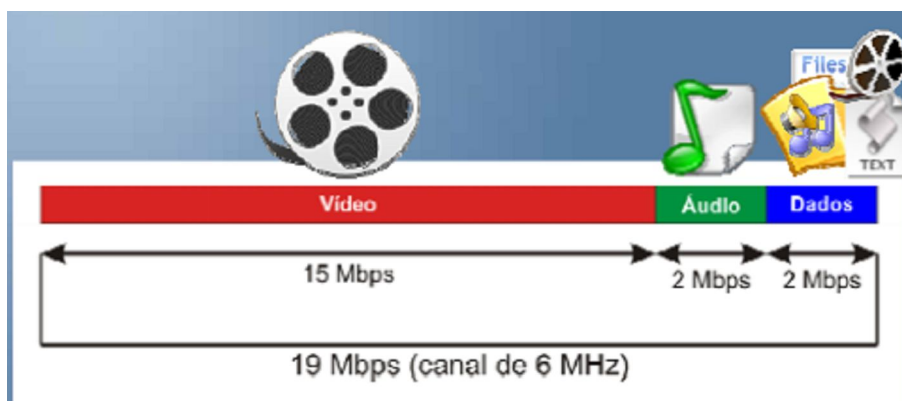


Figura 14 – Divisão da Banda 19Mbps TV Digital

O assunto transmissão de dados traz obrigatoriamente um método conhecido como Modulação, que é um processo na qual a informação é adicionada a ondas eletromagnéticas. É assim que qualquer tipo de informação (digital ou analógica), até a voz humana ou transação de dados numa aplicação interativa é transmitida numa onda eletromagnética. O transmissor adiciona a informação numa onda básica de tal forma que poderá ser recuperada na outra parte através de um processo reverso chamado Demodulação.

A modulação digital é o processo que possibilita alteração de característica(s) de um sinal analógico de acordo com a informação digital a ser transmitida. De forma geral, tais características são:

- ✓ Amplitude;
- ✓ Fase;
- ✓ Frequência.

A seguir são apresentadas as técnicas ASK (amplitude), FSK (frequência) e PSK (fase) que são utilizadas para transmissão analógica de dados digitais:

✓ ASK: *Amplitude Shift Keying*

Uso da alteração na amplitude do sinal para a transmissão de dados. Basicamente, a transmissão do bit 1 se dá pela transmissão da portadora e a transmissão do bit 0 se dá pela redução da amplitude da portadora. Aplica-se tal técnica em modems, pois sinais digitais precisam ser condicionados caso contrário serão distorcidos na linha de transmissão, o que acarretará em perda. Nos modems atuais, entretanto, utilizam-se técnicas de modulação mais avançadas que o ASK, obtendo assim taxas de transmissão maiores.

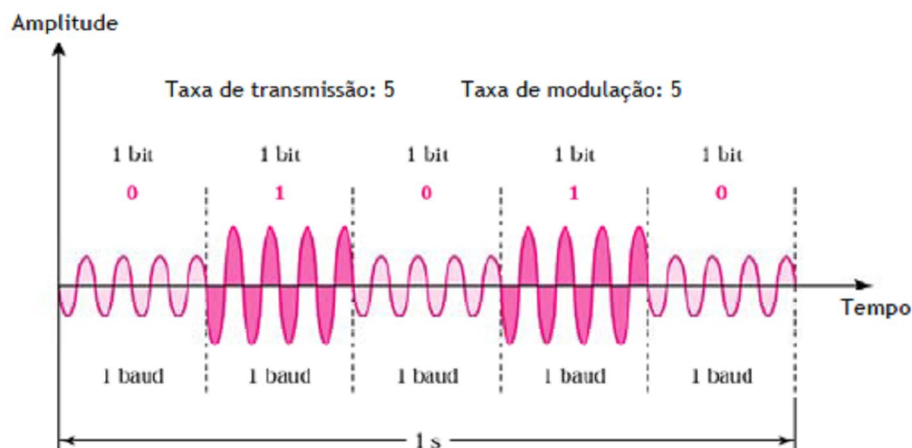


Figura 15 – Modulação ASK

✓ FSK: *Frequency Shift Keying*

Uso da alteração na frequência do sinal para a transmissão de dados. Basicamente, na transmissão de um bit 1 se transmite uma portadora numa determinada frequência e a transmissão do bit 0 se transmite uma portadora em outra frequência.

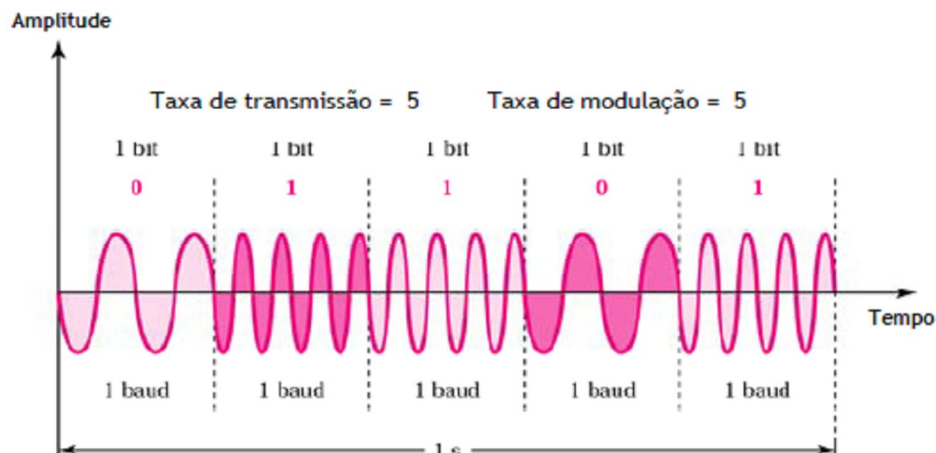


Figura 16 – Modulação FSK

✓ PSK: *Phase Shift Keying*

Uso da alteração na fase do sinal para a transmissão de dados Basicamente, na transmissão de um bit 1 se transmite uma portadora numa determinada fase e a transmissão do bit 0 se transmite uma portadora em outra fase. Abaixo, vemos o 2-PSK.

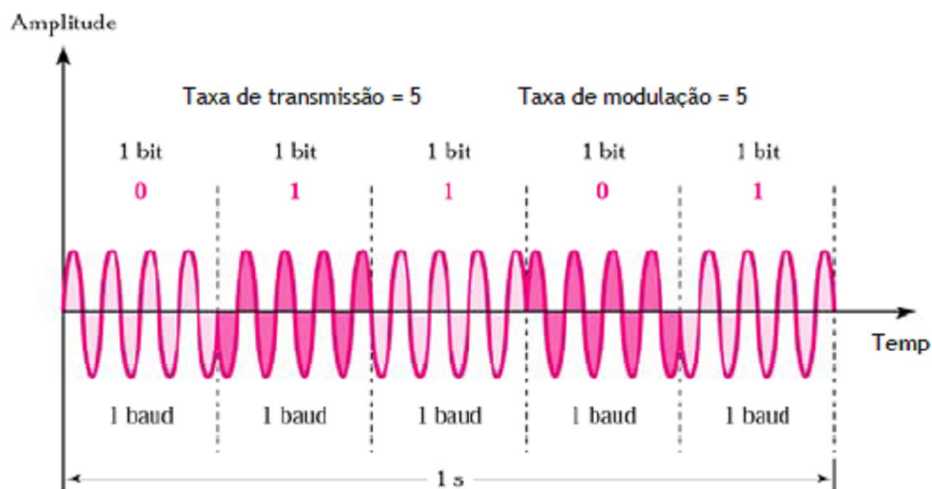


Figura 17 – Modulação 2-PSK

A técnica PSK acima utiliza apenas duas fases 0o (para o bit 0) e 180° para o bit 1. Contudo, podemos utilizar mais fases visando aperfeiçoar a transmissão. Esse método abaixo, representando dois bits (Dibit) é conhecido como 4-PSK.

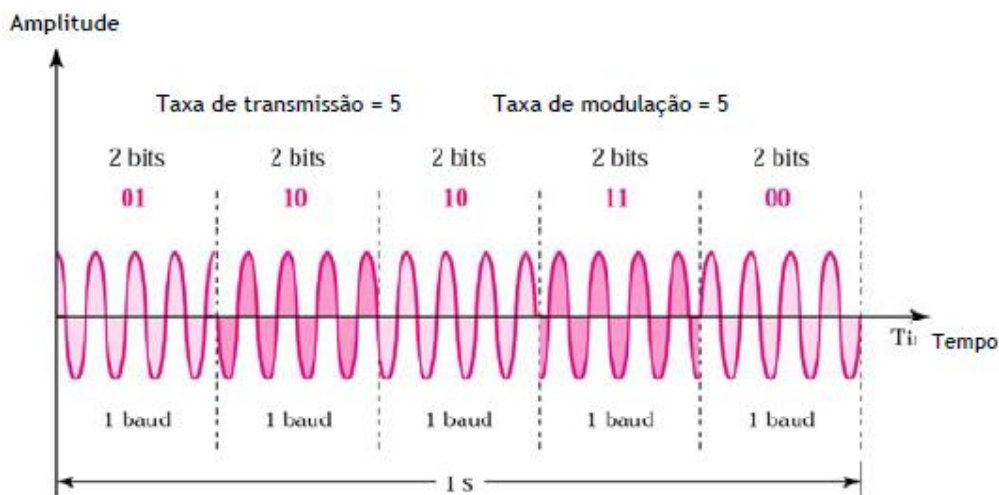


Figura 18 – Modulação 4-PSK

Note que no gráfico anterior são utilizadas quatro fases distintas:

Tabela 03 – Fases distintas para a modulação 4-PSK (Fonte: Bezerra, 2008)

FASE	BITS
0°	00
90°	01
180°	10
270°	11

Visando aumentar mais a transmissão de bits por segundo, foi criada a técnica QAM (Quadrature Amplitude Modulation). Técnica para codificar dados digitais em um sinal analógico através de modulação em que duas componentes diferentes são combinadas em um único sinal através de modulação ortogonal destas duas componentes, evitando assim a interferência. Daí o termo "quadratura".

A técnica empregada consiste na combinação da modulação por amplitude (AM) com modulação por fase (PSK) para criar uma constelação de pontos de sinal, cada qual representando uma combinação exclusiva de bits. Utilizada em TV digital e outros sistemas que necessitam de alta taxa de transferência de informação. Na imagem abaixo vejam o domínio do tempo para o sinal 8-QAM, com taxa de transmissão de 24 e taxa de modulação de 8.

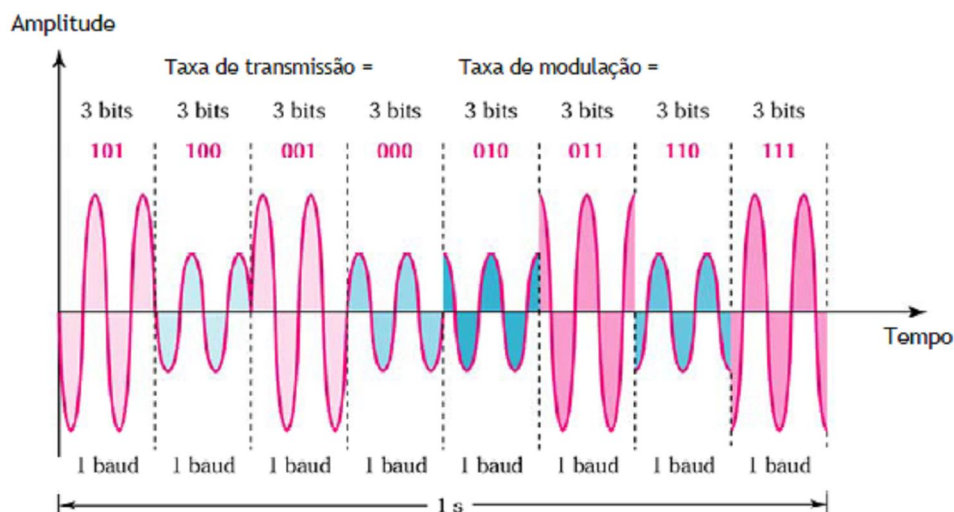


Figura 19 – Modulação 8-QAM

A seguir, as constelações de 4-QAM e 8-QAM:

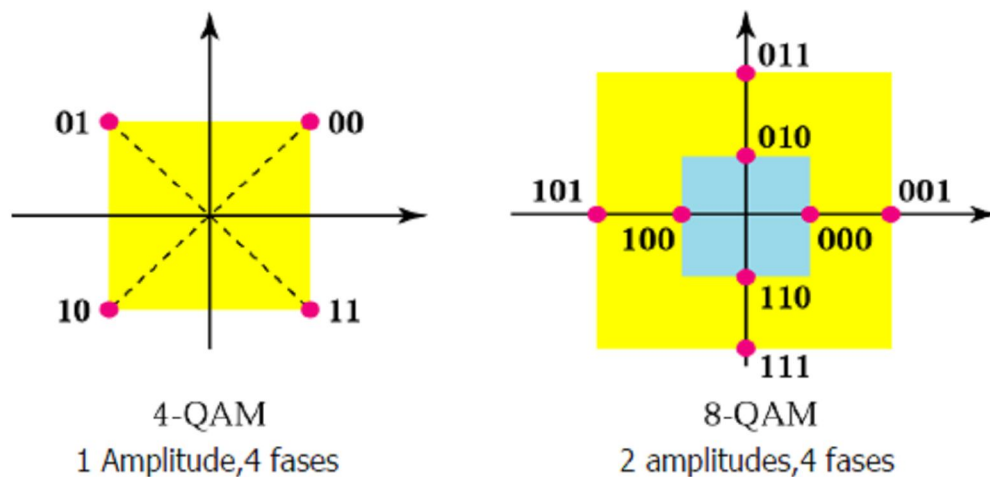


Figura 20 – Constelações 4-QAM e 8-QAM

Existem diversas combinações possíveis para criar uma constelação, cada uma com sua particularidade em relação à outra, mas de forma geral a QAM possui menos interferência à ruídos que a ASK e melhor utilização de banda que a PSK.

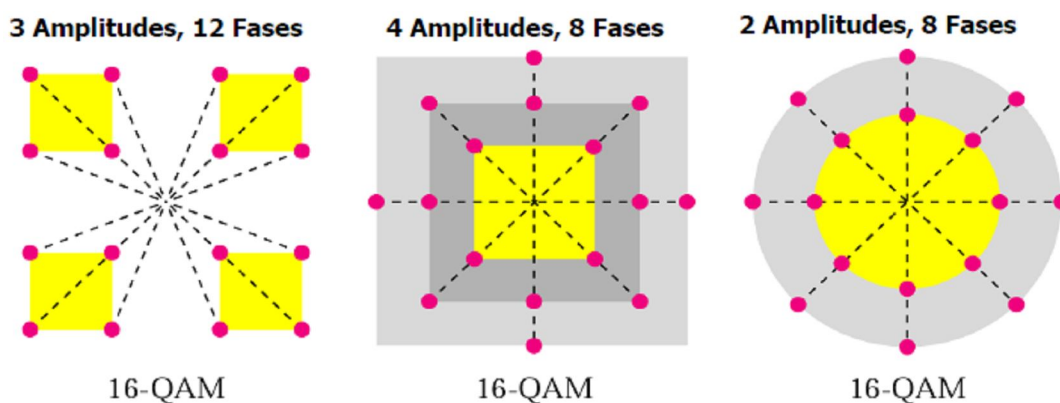


Figura 21 – Variações das constelações 16-QAM

### 3.8. Imagem e Padrões de cor

Todas as imagens digitais são constituídas por um mesmo elemento no seu nível mais básico: o pixel. A palavra vem da junção das palavras "*PIC*tura" e "*EL*ement", que significam "imagem" e "elemento", respectivamente. É o menor elemento num dispositivo de exibição, ao qual é possível atribuir-se uma cor. Cada imagem é composta por milhares de pixels. Algumas definições mais técnicas para o pixel, traz sendo como *pontos de luz*, já que cada pixel é formado por 03 micro pontos de luz (01 vermelho, 01 verde e 01 azul), formando um triângulo. Cada conjunto desse é um pixel, o ponto de luz (ou ponto de cor) que "vemos" a olho nu no monitor. Quando falamos que uma imagem tem 800 x 600 pixels (largura x altura), por exemplo, significa que ela possui 800 pontos de luz vistos a olho nu em cada linha, e 600 pontos de luz vistos a olho nu em cada coluna.

O que é chamado de luz faz parte da faixa de frequência eletromagnética. Chamamos de luz a parte visível dessas radiações, que fica entre as frequências infravermelhas e ultravioletas. Na faixa visível temos aproximadamente cerca de 600.000 matizes de cor, e todas somadas formam a luz branca. Abaixo, na figura 22, vemos o espectro eletromagnético, que mostra exatamente a faixa de frequência que conseguimos captar e enxergar. (Fonte: Tudo sobre TV, 2010).

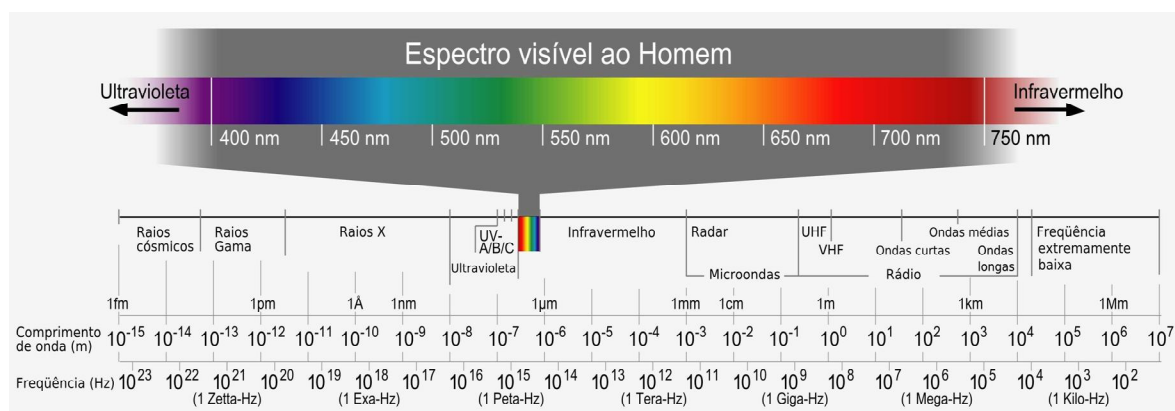


Figura 22 – Espectro Eletromagnético (Fonte: SBFísica,2010)

As imagens que se formam no receptor de televisão consistem de uma série de elementos tendo diferentes quantidades de luz e sombra com uma distribuição não uniforme nas linhas horizontais e através dos campos verticais. Numa cena real, o sinal de vídeo conterá uma sucessão de tensões se modificando continuamente dentro de cada linha para diferentes

pixels. Durante a varredura, é varrido um campo de linhas ímpares e depois um campo de linhas pares, formando assim, um quadro. Graças à persistência na retina do olho humano, a sucessão de quadros é interpretada pelo cérebro como uma imagem contínua.

Através do tamanho da imagem, temos a relação de aspecto (aspect ratio), que significa a relação entre a largura e a altura da imagem. Nos sistemas SDTV – *Standard Definition Television* - transmissões que não são em alta definição, temos o aspect ratio de 4:3 (4 de largura por 3 de altura).

1	2	3	4
2			
3			

Figura 23 – Aspecto 4:3  
(Fonte: Wordpress, Blog Armando Moraes, 2009)

Já nos sistemas EDTV - *Enhanced Definition Television* - com definição equivalente ao SDTV porém em formato retangular, widescreen, e no HDTV – *High Definition Television* – transmissões em alta definição, temos 16:9 (16 de largura por 9 de altura). Há também, no caso do modelo brasileiro, a transmissão em *Low Definition Television* (LDTV), com 320x240 linhas, destinada a recepção em celulares, aparelhos de TV portáteis e computadores.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2															
3															
4															
5															
6															
7															
8															
9															

Figura 24 – Aspecto 16:9  
(Fonte: Wordpress, Blog Armando Moraes, 2009)

Portanto imagens produzidas em HD têm proporcionalmente mais elementos na largura do que as produzidas em SD.



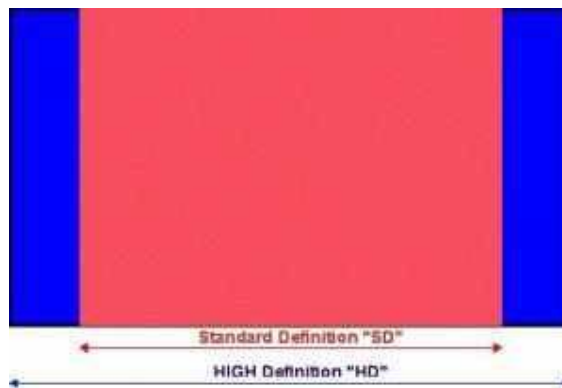


Figura 25 – HD x SD  
(Fonte: Wordpress, Blog Armando Moraes, 2009)

O progresso da televisão pode ser avaliado em função do número de linhas de que é composta a imagem transmitida. As primeiras imagens de televisão possuíam 24 ou 48 linhas; depois esse número foi gradativamente aumentando para 60, 120, 180, 240, 343, 441, e finalmente chegou ao padrão atual de 525 linhas, quando a rede elétrica é de 60Hz, ou de 625 linhas para redes elétricas de 50Hz.

Dessa forma, conseguimos diferenciar os padrões de cor existentes no mundo. Basicamente são três: PAL, NTSC e SECAM. (Fonte: Fazendo Vídeo, 2010).

✓ PAL: abreviatura do inglês *Phase Alternating Line* (*Linha de Fase Alternante*) é o formato mais comum na Europa. Foi criado pelo engenheiro alemão Walter Brunch, juntamente com a empresa eletrônica Telefunken. Tem uma melhor qualidade de imagem, já que dispõe de um maior número de linhas de varrimento. O sistema PAL apresenta mais 100 linhas de resolução, comparativamente ao sistema NTSC, ou seja, 625 linhas de resolução. Em contra-partida, e para fazer face ao tipo de corrente alternada que é usada na Europa, de 50Hz (50 ciclos por segundo), tem varredura de 50 meios-frames por segundo, ou mais exactamente, 25 frames por segundo. Variantes do sistema PAL: PAL-B, PAL-G, PAL-I, PAL-N, PAL-M – sistema utilizado no Brasil desde 19 de Fevereiro de 1972. Foi criado para fazer face à rede elétrica de 60 Hz;

✓ NTSC: abreviatura de *National Television Systems Committee*, a comissão que desenvolveu os padrões de televisão para os Estados Unidos. Apresenta 525 linhas de resolução, das quais 480 são a imagem visível e as outras 45 são as VBI (*do inglês – Vertical Blanking Interval*), que servem para estabelecer o tempo que o feixe de elétrons necessita para se reposicionar, do final de um frame para o início

do próximo frame. Este formato de vídeo transmite 60 meios-frames por segundo (mais exactamente 59,94), ou seja, apresenta 29,97 fps (ou quadros) por segundo. A varredura é por isso, intercalada. Tem uma relação de aspecto de 4:3. Com este sistema de vídeo, conseguem-se obter até 16 milhões de cores diferentes.

✓ SECAM: abreviatura de *Séquentiel Couleur à Mémoire*, ou em Português Cor Sequencial com Memória, é um sistema a cores analógico, que apareceu na França, em 1956. Tal como no sistema PAL, também o sistema SECAM transmite 625 linhas de resolução, das quais 576 são visíveis. Também utiliza a frequência de varredura de 50Hz. Hoje em dia, muitos dos trabalhos são realizados em sistemas de vídeo que suportem 576 linhas de resolução (PAL e SECAM) ou de vídeo composto (PAL), sendo depois convertidos para SECAM para a transmissão.



Figura 26 – Padrões de cor no mundo (Fonte: Wikipédia, 2010)

Outra forma de classificar uma imagem é em relação ao tipo de varredura, que pode ser entrelaçada ou progressiva. Na varredura entrelaçada, a tela é preenchida em duas fases: na primeira fase as linhas ímpares são exibidas e depois, as pares. Na varredura progressiva todas as linhas são exibidas ao mesmo tempo. (Televisão Digital, 1ª Ed, Bastos e Fernandes).

Então, quando uma imagem é classificada como 720p, quer dizer que tem 720 linhas horizontais e varredura progressiva. Outro exemplo seria uma imagem 1080i, 1080 linhas horizontais e varredura entrelaçada, esta é a definição máxima prevista para a transmissão de sinal digital no modelo do SBTVD (ABNT, 2007a). Já a definição de 1080p só é encontrada em *videogames* e aparelhos de *blu-ray*.

### 3.9. Áudio

Para os humanos, a audição é limitada por frequências entre 20Hz e 20.000 Hz (20 kHz), embora estes limites não sejam absolutos. Valores abaixo de 20Hz são os infra-sons; e valores acima de 20KHz, os ultra-sons. Ambos não são captados por nós.

As mudanças de qualidade na TV Digital não se restringem só ao vídeo e inserções de dados para interação. O áudio é outro ponto de extrema importância que deve ser também estudado. De nada adianta um vídeo de excelente qualidade, se o áudio também não vier acompanhando todo o impacto visual, dando total qualidade à cena final.

Na TV analógica utiliza-se áudio mono ou estéreo. Este último é mais nítido, limpo e é utilizado 2 canais (direito e esquerdo). Na TV digital utilizam-se multicanais. No SBTVD é utilizado o formato MPEG-4 ACC com 5.1 canais, ou seja, são 05 caixas cada uma saindo um áudio diferente, o que proporciona para o telespectador um som de altíssima qualidade, com uma sensação de estar assistindo a um filme no cinema.

Multicanais traz uma combinação de até 06 canais, com o intuito de tornar o som espacial:

1. Um canal Esquerdo Frontal (Lf);
2. Um canal Direito Frontal (Rf);
3. Um canal Central (C);
4. Um canal Esquerdo Envolvente Traseiro (Ls);
5. Um canal Direito Envolvente Traseiro (Rs);
6. Um canal de baixas frequências (LFE).

O áudio analógico usado atualmente ainda terá sua decodificação feita pelos STB (set-top-box), porém com saída estéreo, via conectores RCA. Ou seja, nesta fase que nos encontramos, de implantação da TV Digital, serão usados dois canais, que darão a sensação espacial (surround) de acordo com os recursos da TV ou do Receiver. (Fonte: TV Globo Digital, 2010).

### 3.10. Datacasting

O *Data Broadcasting*, ou simplesmente *datacasting*, pode ser definido como um serviço, diferente do serviço de difusão normal, de entrega de informação (seja ela no formato de dados, texto, som, imagem ou em qualquer outro) para clientes com terminais de acesso adequados, através do mesmo meio utilizado na difusão. Dessa forma, mostra-se como a base para o surgimento de novos serviços na Televisão Digital.

Existem basicamente quatro mecanismos, ou áreas de aplicação, para a difusão de dados utilizando os fluxos de transporte MPEG-2 na televisão digital: (Fonte: Revista da SET, Ed.86).

✓ *Data Piping*: Canalização de dados. É o método mais simples de inserção de dados em um fluxo de transporte. Baseia-se em encapsular dados brutos diretamente nas cargas de pacotes de transporte;

✓ *Data Streaming*: Fluxo de dados. Pode ser definido como uma área de *datacasting* onde dados, na maioria das vezes *não delimitados*, são continuamente difundidos e geralmente alimentam alguma aplicação do terminal de acesso;

✓ *MPE: Multiprotocol Encapsulation*. É utilizado para transportar datagramas de diversos protocolos através de seções privadas do padrão MPEG-2. O padrão DSM-CC estende as seções privadas, definindo a chamada *seção de datagrama*. Uma *seção de datagrama* pode ser codificada de acordo com qualquer tipo de protocolo de rede da terceira camada do modelo ISO/OSI;

✓ Carrosséis: Esse item será melhor explorado no item 3.14.1.17.

### 3.11. Recepção Móvel (1-SEG)

A possibilidade de recepção de sinais digitais de televisão não fica restrita aos aparelhos comuns. Essa nova tecnologia trouxe uma diversidade no que tange à esse processo, pois de acordo com a ARIB<sup>4</sup> - *Association of Radio Industries and Businesses* – a grande

---

<sup>4</sup> ARIB – *Associação de Indústrias e Empresas de Rádio* - é uma organização de padronização japonesa. O ARIB é designado como o "centro da promoção do uso eficiente do espectro de rádio" pelo Ministro

novidade é que é possível receber o mesmo sinal por aparelhos portáteis como celulares, notebooks, palm tops e televisões portáteis.

A televisão móvel e portátil traz meios de acesso e inclui na audiência pessoas em trânsito, que atualmente, no máximo, escutam rádio. No primeiro caso, onde o sinal pode ser recebido por meio de antenas e aparelhos de TV móveis, a configuração da audiência não difere muito na forma de ver TV na sala de estar. No entanto, a recepção portátil, seja com aparelhos portáteis ou via aparelhos celulares, altera a forma de consumir e enxergar o conteúdo da TV. Primeiramente, o formato, a resolução e a definição das telas mudam; segundo, a demanda de informações difere, pois o tempo de audiência varia muito em relação à audiência da sala de TV; terceiro, a possibilidade de interatividade plena ou permanente é real, pois o canal de retorno é intrínseco ao meio da telefonia celular.

No padrão japonês e brasileiro, para a recepção via celular não é necessário que o sinal seja habilitado pela operadora de telefonia, pois o sinal é captado diretamente pelo *hardware* do telefone. Não existe tarifa cobrada do usuário do telefone por assistir TV pelo celular, o que não acontece em outros países onde a operadora cobra uma taxa pela utilização do serviço.

Em junho de 2010, a LG foi a primeira fabricante de celular no país a lançar um aparelho com o Ginga instalado. O LG TV Phone GM600 permite interação com a programação da TV Globo, parceira inicial, para a Copa do Mundo.

### **3.12. Arquitetura**

Trataremos aqui da arquitetura do Sistema de Televisão Digital como um todo, visando mostrar os principais elementos do sistema, e suas interações.

A arquitetura que representa as camadas de tecnologia existentes em TV Digital Interativa no mundo é apresentada na figura 27. O modelo de arquitetura do SBTVD, com seus padrões de referência, que são um conjunto de padrões que regulam cada um dos procedimentos descritos, é mostrado na figura 28.

---

das Relações Internas e Comunicações. Formado por um grupo de empresas, fabricantes e operadoras de televisão e de telecomunicações, fomentado pelo governo japonês.

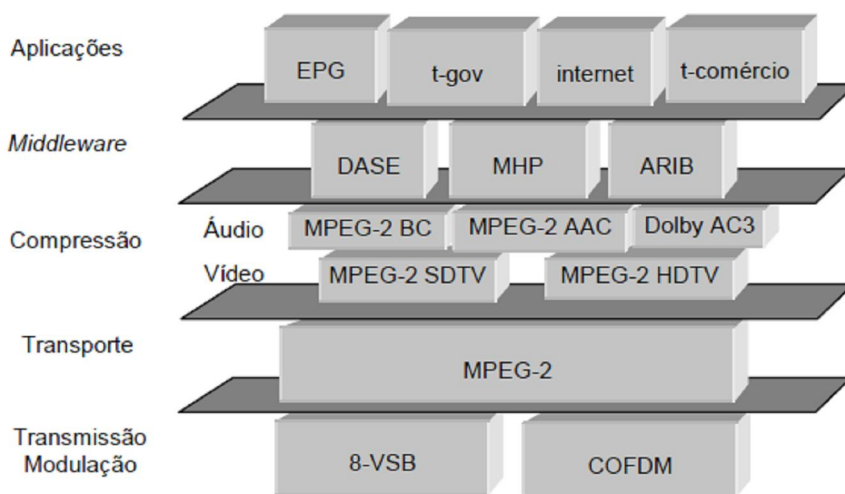


Figura 27 – Arquitetura da TV Digital (Fonte: Grupo Ginga Goiás, 2010)

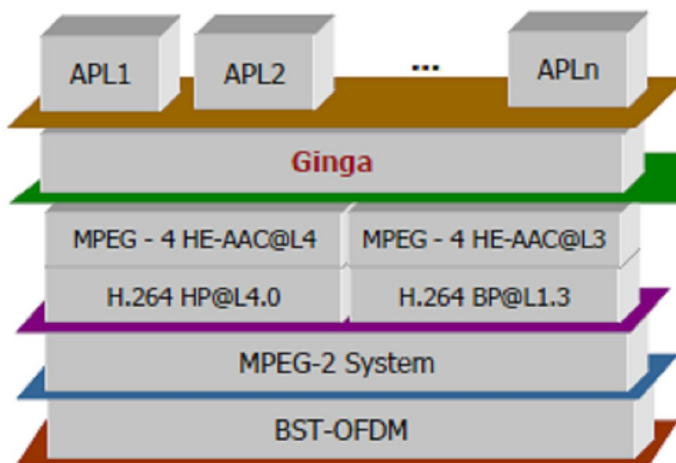


Figura 28 - Padrões de Referência do SBTVD (Fonte: Soares e Barbosa, 2009)

A idéia por detrás da arquitetura é a de que cada camada oferece serviços para a camada superior, e usa os serviços oferecidos pela camada inferior. Dessa forma, uma aplicação que executa em TV digital interativa faz uso de uma camada de *middleware*, que intermedeia toda a comunicação entre a aplicação e o resto dos serviços oferecidos. Na figura 28, temos de baixo para cima: camada de modulação, multiplexação, codificação de vídeo, codificação de áudio, middleware e aplicações.

A finalidade da camada de *middleware* – ou camada do meio – é oferecer um serviço padronizado para as aplicações (camada de cima), escondendo as peculiaridades e heterogeneidade das camadas inferiores (tecnologias de compressão, de transporte e de

modulação). O uso de *middleware* facilita a portabilidade de aplicações, que podem ser transportadas para qualquer receptor digital (ou *set-top Box*) que suporte o *middleware* adotado. Essa portabilidade é primordial em sistemas de TV digital, pois é muito complicado considerar como premissa que todos os receptores digitais sejam exatamente iguais.

As principais especificações existentes de TV digital – norte-americano, europeu e japonês – adotam diferentes padrões para *middleware* em seus receptores digitais. No próximo item, esses padrões de TV digital e seus *middlewares* são comparados. (Fonte: TVDI, 2010).

### **3.13. TV Digital no Mundo**

Atualmente três importantes sistemas abertos para a televisão digital encontram-se em desenvolvimento.

O sistema norte-americano ATSC - *Advanced Television Systems Committee*, o sistema predominantemente europeu DVB - *Digital Video Broadcasting* e o sistema japonês, conhecido como ISDB - *Integrated Services Digital Broadcasting*.

#### **3.13.1 ATSC – Padrão Americano**

Sistema ATSC (Advanced Television Systems Committee): adotado nos Estados Unidos, Canadá, México, Guatemala, Honduras, El Salvador, Bahamas, Coréia do Sul e Porto Rico, produz imagens no formato 16:9 (wide screen) e com até 1920×1080 pixels – seis vezes mais que o padrão analógico que sucedeu o NTSC. Permite transmitir até seis canais virtuais em definição padrão e oferece qualidade de som similar a dois *home theaters*, por meio do sistema Dolby Digital, que utiliza seis canais de áudio. O consórcio existe desde 1982, mas o padrão só entrou em funcionamento comercial nos Estados Unidos em 1998. É considerado o mais robusto, ideal para transmissão em alta-definição, mas é o menos desenvolvido no quesito mobilidade.

### 3.13.1.1. Middleware – Padrão Americano

#### *DASE-ATSC (DTV Application Software Environment Level1)*

O DASE é uma camada de software que permite às aplicações (softwares em geral) e a programação a executarem em um equipamento chamado “receptor comum”. Aplicação DASE é a coleção de informações que expressa um conjunto específico de comportamentos observáveis. O DASE foi dividido em 2 partes, como pode ser visto na figura 29:

1) Ambiente de Aplicações Declarativas: é basicamente o browser de documentos multimídia (User Agent). Entende-se por Aplicações Declarativas como o documento multimídia composto por regras de estilo, scripts, markups, gráficos, vídeo e áudio.

2) Ambiente de Declarações Procedurais: é a JAVA Virtual Machine (JVM) e a implementação de suas APIs. As Aplicações Procedurais utilizam uma API desenvolvida pela Sun chamada de JavaTV. Os aplicativos desenvolvidos através dessa API são chamados de Xlets, composto por código binário, compilado pela JVM, em conjunto com outros conteúdos como gráficos, vídeos e áudios.

O DASE define ainda as seguintes capacidades:

- ✓ Capacidade de entrada para usuários: interação com os usuários através de navegação (o mesmo quando você navega na web);
- ✓ Capacidade de áudio: decodificação em tempo real e apresentação do conteúdo do fluxo de áudio;
- ✓ Capacidade de vídeo: decodificação de vídeo em tempo real e apresentação do conteúdo do fluxo de vídeo;
- ✓ Capacidade gráfica: decodificação e a apresentação de conteúdo visual, que não seja vídeo, de acordo com várias resoluções;
- ✓ Modelo de Display: utiliza modelo baseado em planos de apresentação (display plan).



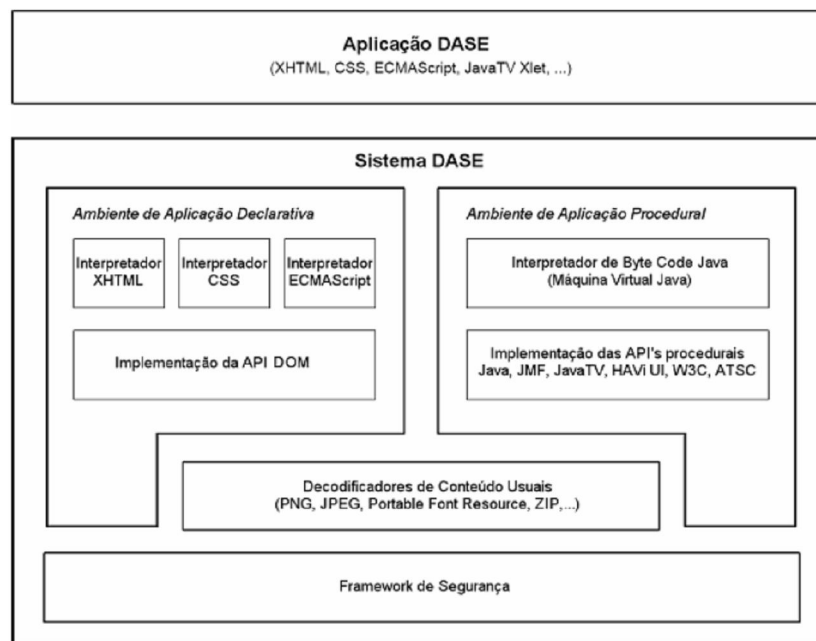


Figura 29 – Middleware Americano – DASE (Fonte: ATSC, 2009)

### 3.13.2 DVB – Padrão Europeu

DVB (Digital Video Broadcasting): adotado comercialmente em 1998, pelo Reino Unido, o padrão também foi abraçado por Índia, Austrália e Nova Zelândia. O consórcio responsável pela sua definição reúne mais de 270 empresas. Possui padrões para transmissão terrestre (DVB-T), por cabo (DVB-C) e satélite (DVB-S). É conhecido por ser mais versátil, facilitando a transmissão de múltiplos canais virtuais na mesma frequência. Opera na frequência de 8 MHz, fator que o deixa em desvantagem em relação ao japonês e ao americano, que operam em 6 MHz, mesmo espectro usado no Brasil para a TV aberta.

A transmissão digital já foi implementada em 21 países membros da comunidade europeia (Áustria, Bélgica, Bulgária, República Tcheca, Dinamarca, Alemanha, Estônia, Grécia, Espanha, França, Hungria, Itália, Lituânia, Luxemburgo, Malta, Países Baixos, Letônia, Finlândia, Suécia, Eslovênia e Reino Unido).

Outros cinco países (Chipre, Irlanda, Polônia, Portugal e Eslováquia) anunciaram que darão início do serviço de TV digital até o final de 2010. A Romênia ainda não fez nenhum anúncio quanto a essa questão. (Fonte: Teleco,2010)

Em relação a recepção de TV, esse padrão exige um canal adicional somente para a transmissão de TV portátil. Isto gera a necessidade de licitações adicionais de canais no

espectro e, conseqüentemente, impõe a tarifação ao consumidor final, uma vez que não existe receita publicitária para arcar com os custos de transmissão, como na TV aberta tradicional.

#### 3.13.2.1. Middleware – Padrão Europeu

MHP (Multimídia Home Platform): O middleware europeu prioriza a segurança e interoperabilidade. Foi desenvolvido para suportar um grande número de serviços, entre eles o Web Browsing. Uma característica interessante do MHP é que, assim como o GINGA, ele é de código aberto.

Por essa característica, o MHP, figura 30, suporta diversos tipos de serviços e é executado em diversas plataformas de hardware, podendo receber o sinal de várias operadoras de TV Digital.

Abaixo são apresentados alguns tipos de transmissão suportados pelo middleware:

- ✓ Enhanced Broadcast: Combina transmissão de áudio e vídeo e serviços de download de aplicações, que permitem serviços de interação local. Este tipo de configuração não suporta canal de retorno. Suporte a linguagem HTML pode ser incrementado a partir de plug-in;
- ✓ Interactive Broadcast: Contém todas as funcionalidades do Enhanced e também permite diferentes formas de interação, modo Global, com ou sem associação com serviços de Broadcast. Esta configuração requer canal de retorno e suporta Internet Protocol (IP);
- ✓ Internet Access: Possui todas as funcionalidades dos modos antecessores e permite acesso a serviços Internet. Interação entre serviços Broadcast e serviços Internet são possíveis. Browser para e-mail e API Java para acesso à Internet também são comuns nesta configuração.

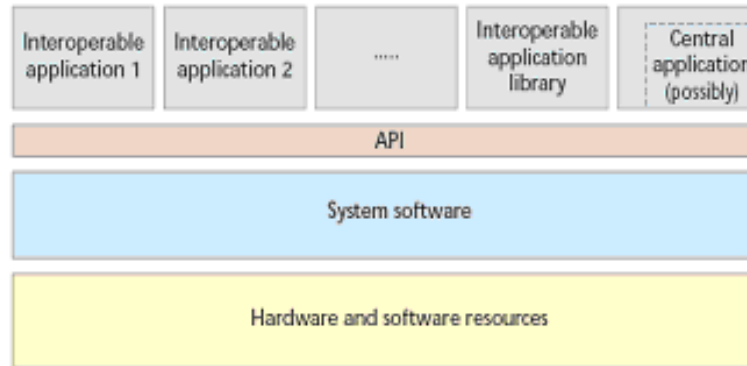


Figura 30 – Middleware Europeu – DVB (Fonte: “TV Interactiva”, 2009)

### 3.13.3 ISDB – Padrão Japonês

ISDB (Integrated Service Digital Broadcasting): vem sendo desenvolvido desde a década de 70, mas só entrou em operação em 2003, na região de Tóquio, no Japão. É uma evolução do DVB-T, ao qual foram acrescentadas as seguintes implementações:

- ✓ Foi acrescentado um “Interleaver” temporal para melhorar o desempenho na presença de interferências concentradas, tais como o ruído impulsivo;
- ✓ A banda de RF de 6MHz foi subdividida em 13 segmentos independentes, com a possibilidade de serem enviadas 3 programações diferentes ao mesmo tempo, por exemplo: uma em QPSK, outra em 16QAM e outra em 64QAM;
- ✓ Foi acrescentado o modo 4K;
- ✓ Foi acrescentado o método de modulação DQPSK “Differential Quaternary Phase Shift Keying”.

#### 3.13.3.1. Middleware – Padrão Japonês

ARIB (Association of Radio Industries and Business): O padrão ARIB define as regras de aplicação de um modelo de referência para o serviço de broadcasting de dados, transportado como parte do serviço de broadcasting digital, definido pelo padrão Japonês de broadcasting digital. Neste sistema, áudio, vídeo e todos os serviços de dados são multiplexados e transmitidos via broadcasting de rádio, em um fluxo empacotado (Transport Stream – TS), especificado pelo MPEG-2.

Canais para a interatividade das comunicações são disponibilizados através dos canais interativos da rede, tanto fixas quanto móveis.

Três tipos de sistemas de transmissão de dados são suportados pelo ARIB:

- ✓ Transmissão de dados que utiliza o armazenamento dos pacotes como um fluxo de pacotes no PES (Packetized Elementary Stream);
- ✓ Transmissão de dados que utiliza as seções, utilizado para serviços de armazenagem de informação (data storage services);
- ✓ Sistema onde os dados são armazenados diretamente no payload do pacote TS.

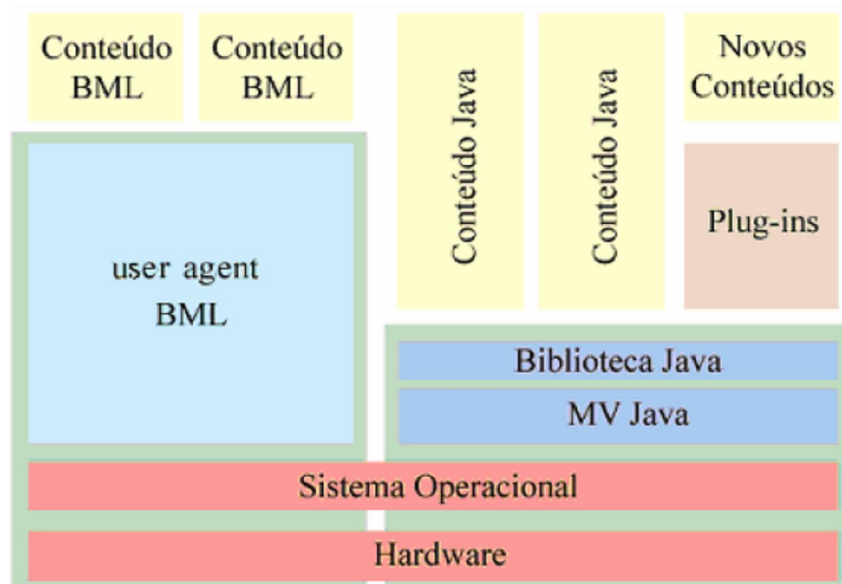


Figura 31 – Middleware Japonês – ARIB (Fonte: NHK, 2009)

A tabela a seguir faz um comparativo entre os três sistemas com relação aos padrões e características de cada um a respeito da difusão terrestre.

Tabela 04 – Características dos três sistemas abertos com relação à difusão terrestre (Fonte: Piccioni, 2005)

Sistema	DVB	ATSC	ISDB
Modulação	COFDM	8-VSB	COFDM
Multiplexação	MPEG-2 Sistemas	MPEG-2 Sistemas	MPEG-2 Sistemas
Codificação de Vídeo	MPEG-2 Video	MPEG-2 Video	MPEG-2 Video
Codificação de Áudio	MPEG-2 Layer II	Dolby AC3	MPEG-2 AAC
Middleware	MHP	DASE	ARIB-Std-B24
Largura de Banda	6 a 8 MHz	6 a 8 MHz	6 a 8 MHz
Taxa de Transmissão	De 4,98 à 31,67 Mbps	19,4 Mbps	De 3,65 à 23,23 Mbps

Cada sistema possui seus méritos e limitações devido às escolhas feitas, levando-se em conta fatores técnicos, econômicos e aplicabilidade. Devido as suas diferenças torna-se possível uma avaliação quanto aos desempenhos quando confrontados um a um, principalmente em etapas tão importantes para o sucesso da TV digital como os subsistemas de codificação e modulação.

Abaixo, temos uma figura demonstrando os sistemas já usados nos países. A legenda encontra-se na tabela 05.

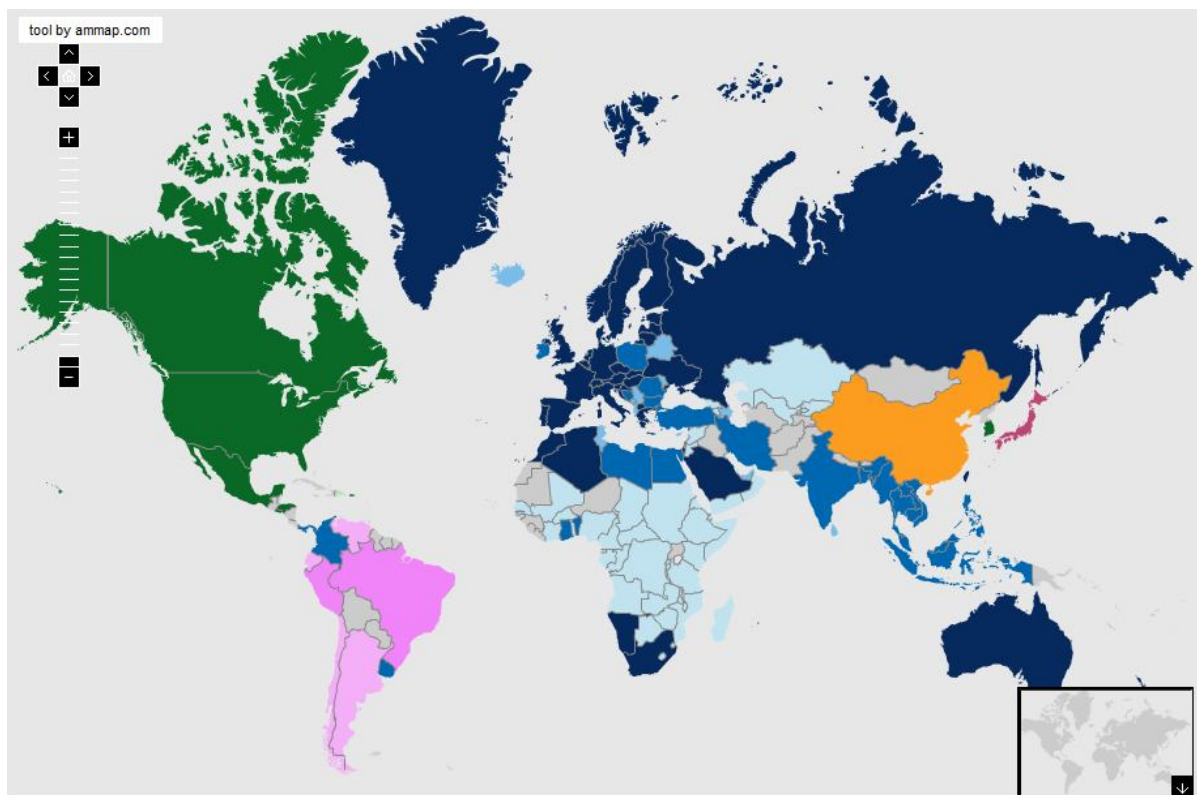


Figura 32 – Mapa dos Sistemas de TV Digital no Mundo (Fonte: DTVStatus, 2010)

Tabela 05 – Legenda do Mapa TV Digital no Mundo (Fonte: DTVStatus, 2010)

Sistema	Explicação
DVB-T	Transmissão via DVB-T que está ativamente em uso.
DVB-T adotado	Países que adotaram o sistema DVB-T.
DVB-T em fase de teste	Países que estão em fase de testes com o DVB-T.
RRC06	Países que participaram da Conferência Regional de Radiocomunicações de 2006 da ITU. Pode-se supor que todos os países que participaram usarão o sistema DVB-T quando se migrarem do analógico para o digital.
ATSC	Rádiodifusão através do sistema ATSC que está ativamente em uso.
ATSC adotado	Países que adotaram o sistema ATSC.
ATSC em fase de teste	Países que estão em fase de testes com o ATSC.
ISDB-T	Rádiodifusão via ISDB-T que está ativamente em uso.
ISDB-T adotado	Países que adotaram o sistema ISDB-T.
ISDB-T em fase de teste	Países que estão em fase de testes com o ISDB-T.
SBTVD-T	Rádiodifusão via SBTVD-T que está ativamente em uso.
SBTVD-T adotado	Países que adotaram o sistema SBTVD-T.
DMB-T / H <sup>5</sup>	Rádiodifusão via DMB-T / H que está ativamente em uso.
DMB-T / H adotado	Os países que adotaram o DMB-T / H
DMB-T / H em fase de teste	Os países comprometem-se ensaios com DMB-T / H.
Os serviços comerciais DVB-T	Nenhuma aprovação formal de uma norma da TDT.
	países indecisos.

<sup>5</sup> O DMB-T/H é o padrão de televisão digital criado e adotado na República Popular da China. Adotado na China, em Hong Kong e Macau. Seus primeiros testes foram durante as Olimpíadas de Pequim. Ainda se encontra em desenvolvimento o que acaba por atrapalhar a sua expansão. As rivalidades e desconfianças existentes entre as nações asiáticas fez com que elas não adotassem um mesmo padrão de tv digital desenvolvido em comum acordo.

### 3.14. TV Digital no Brasil

A televisão é um dos mais importantes meios de comunicação no Brasil. De acordo com o site Teleco, em 2008, o Brasil ultrapassou o patamar de 95% de domicílios com esse tipo de aparelho, deixando para trás o rádio, telefone e microcomputador. Isso demonstra o enorme interesse da população brasileira com esse tipo de comunicação. Com a TV Digital, a televisão passará a ser também uma ferramenta de inclusão social e digital, além de ser uma evolução tecnológica necessária.

Abaixo, encontra-se uma tabela com os demonstrativos de domicílios e aparelhos de comunicação.

Tabela 06 – Número de domicílios / aparelhos eletrônicos (Teleco, 2008)

Domicílios - % com	2001	2002	2003	2004	2005	2006	2007	2008
Rádio	88,0%	87,9%	87,8%	87,8%	88,0%	87,9%	88,1%	88,90%
Televisão	89,0%	90,0%	90,1%	90,3%	91,4%	93,0%	94,5%	95,10%
Telefone (Fixo ou Celular)	58,9%	61,7%	62,0%	65,4%	71,6%	74,5%	77,0%	82,10%
Microcomputador	12,6%	14,2%	15,3%	16,3%	18,6%	22,1%	26,6%	31,2%
Microcomputador com acesso à Internet	8,60%	10,3%	11,5%	12,2%	13,7%	16,9%	20,2%	23,8%
Total de Domicílios (milhares)	46.507	48.036	49.712	51.753	53.114	54.610	55.770	57.557

Passados pouco mais de dois anos desde a implantação do SBTVD, a realidade no Brasil é a seguinte: 29 cidades brasileiras já transmitem TV Digital em alta definição. São elas: São Paulo (SP), Belo Horizonte (MG), Rio de Janeiro (RJ), Goiânia (GO), Porto Alegre (RS), Curitiba (PR), Campinas (SP), Cuiabá (MT), Salvador (BA), Florianópolis (SC), Vitória (ES), Uberlândia (MG), São José do Rio Preto (SP), Teresina (PI), Santos (SP) Brasília (DF), Campo Grande (MS), Fortaleza (CE), Recife (PE), João Pessoa (PB), Sorocaba (SP), Belém (PA), Manaus (AM), Ribeirão Preto (SP), Joinville (SC), Aracaju (SE), Londrina (PR), São Luís (MA) e Araraquara (SP).



Figura 33 – Estados com TV Digital (DTV,2010)

Especificamente, em Brasília, temos os seguintes canais DTV autorizados:

Tabela 07 – Canais TV Digital em Brasília (Fonte: Própria autora)

Canal Digital	Emissora
15 UHF	EBC
21 UHF	TV GLOBO
23 UHF	TV RECORD
24 UHF	SBT
26 UHF	TV BANDEIRANTE
50 UHF	TV SENADO
52 UHF	TV JUSTIÇA
61 UHF	TV CÂMARA

### 3.15. Sistema Brasileiro de TV Digital (SBTVD)

O SBTVD entrou em operação comercial no dia 02 de dezembro de 2007 na cidade de São Paulo. Após vários anos de estudos dentre os sistemas disponíveis no mundo, chegou-se um consenso que o nosso sistema teria que ter o sistema japonês como base, por ser o mais



adequado para a nossa realidade socioeconômica, entretanto com algumas modificações. Dessa forma, o nosso padrão ISDB-T<sub>B</sub> - apenas esse último “B” diferencia o nome do padrão japonês com o significado de “Brasil” para contemplar essas atualizações feitas – mostrou-se um padrão altamente inovador, inclusive com a sua internacionalização, fabricantes estrangeiros de equipamentos para radiodifusão já anunciaram que terão em breve soluções de transmissão na versão brasileira do padrão, como é o caso da Thomson e a Harris; esta última prepara inclusive uma fábrica no Brasil. (Fórum SBTVD/TelaViva/2010).

De acordo com o Fórum do SBTVD, temos o melhor sistema de TV Digital do mundo:

*“Com esses recursos, uma autoridade japonesa, que estava presente na inauguração da TV Digital Brasileira (02/12/2007, em São Paulo) fez o seguinte comentário: ‘O Brasil melhorou o nosso sistema de TV Digital’. Como o sistema deles já era o melhor dos três, podemos afirmar que temos o melhor sistema de TV Digital do mundo.”*

As atualizações incorporadas ao sistema japonês foram:

- ✓ Para a codificação de áudio e vídeo, usa-se o H.264, também conhecido de MPEG4 parte 10 – para vídeo e codificação HE-AACv2 para áudio, também conhecido como AAC+ (enquanto no Japão usa-se o MPEG2 – vídeo – e MPEG2 Layer3 - áudio);

- ✓ As codificações para recepção móvel (1-SEG) também foram diferenciadas:

- Japão: Vídeo H264 a 15fps; Áudio HE-AAC v.1 low complexity.

- Brasil: Vídeo H264 a 30fps; Áudio HE-AAC v.2 low complexity.

### 3.15.1 Características e especificações técnicas

Dentre as principais características do SBTVD, temos a mobilidade e a portabilidade, a multiprogramação, a interatividade, e a codificação de áudio e vídeo diferenciada. Esse item

traz uma melhor explanação sobre todos esses pontos e outros fatores mais técnicos, abordando inclusive como ocorre todo o processo de transmissão e recepção do sinal digital.

### 3.15.1.1 Mobilidade e Portabilidade

Permite a recepção móvel em celulares, mini-tvs, computadores e veículos em movimento gratuitamente, uma vez que o sinal para os receptores móveis trafega no mesmo canal que a transmissão em alta definição.

O serviço de recepção portátil no sistema ISDB-TB é conhecido como 1SEG. Essa denominação vem da referência ao fato de o sistema de TV digital japonês dividir o sinal em 13 segmentos: 12 segmentos de alta definição e o 13º de baixa definição, próprio para dispositivos portáteis. Este segmento central pode ser submetido ao processo de entrelaçamento de frequência – independente dos outros 12 segmentos. Esse tipo de configuração permite a criação de um serviço portátil (1-segundo), que consiste em uma das camadas do canal de TV. A figura 34 ilustra a divisão do segmento caso for usado apenas com sinais SDTV + 1SEG, surgindo assim a multiprogramação, e HDTV + 1SEG. (Fonte: DTV, 2010).

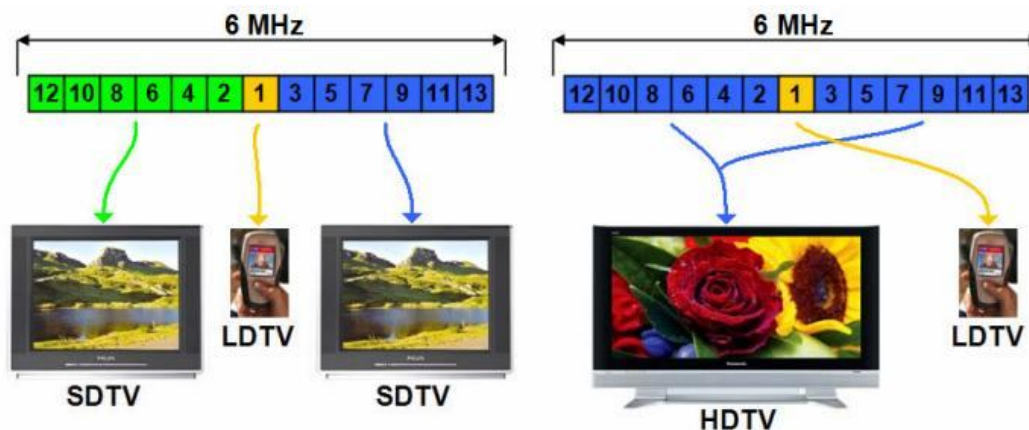


Figura 34 – Segmento 1-SEG (Fonte: Wordpress, Armando Moraes, 2008)

Os padrões de codificação de vídeo e áudio utilizados nas transmissões móveis possuem algumas semelhanças aos utilizados no sistema japonês, mas não são iguais. Para vídeo é usado o H.264 Baseline Profile com resolução de 320x240 pixels (formato 4:3) ou 320x180 pixels (formato 16:9) com bitrate entre 220 a 320kbit/s; para áudio é usado HE-

AACv2 com bitrate entre 48 e 64kbit/s, enquanto no japonês, usa-se o HE-AACv1. (Fonte: Wikipédia, 2010).

Há, porém, modificações feitas para o padrão brasileiro. Enquanto o 1SEG japonês possui taxa de quadros máxima de 15fps para transmissões, o máximo no padrão brasileiro é 30fps. Esse é um dos fatores que impedem que receptores 1SEG japoneses recebam adequadamente transmissões 1SEG brasileiras. O alcance do sinal é de 40 km da antena transmissora, à medida que o sinal 1SEG é muito robusto. O uso no desktop/notebook requer um processamento de no mínimo 1,4ghz, pois ele utiliza muita compressão na transmissão. Apesar de não ser em alta definição, a imagem do 1-seg também é digital, livre de fantasmas, chuviscos e interferências.

Dessa forma, temos as definições: a portabilidade “Permite a recepção em equipamentos portáteis em qualquer localidade, utilizando-se de celulares, televisores de mão ou computadores equipados com receptor de TV estando parado ou em movimento”. A mobilidade é o conceito que “Capta os sinais de TV em dispositivos – celulares e mini-televisão – em movimento, dentro de ônibus, de trens, de carros, entre outros”. (Fonte: DTV, 2010).

### 3.15.1.2 Multiprogramação

Essa característica possibilita às emissoras transmitirem, ao mesmo tempo, mais de um programa, ou imagens em ângulos diferentes de uma atração. Isso é possível, graças à compressão realizada pelo H.264, assim em um mesmo canal de 6MHz poderemos ter a transmissão de até quatro canais em definição standard (SD). Também será possível dividir a tela com imagens em ângulos distintos de um mesmo filme ou jogo de futebol, por exemplo.

Entretanto, o governo federal editou uma norma (*Norma Geral para Execução dos Serviços de Televisão Pública Digital - nº 01/2009, do Ministério das Comunicações*) que proíbe redes comerciais e emissoras públicas estaduais de aderirem à multiprogramação usando o sinal da TV digital. As razões para o Minicom impor essa decisão, são: evitar o aluguel de canais a terceiros (canais de vendas e religiosos), para que não haja uma reprodução, em escala muito maior, do que já acontece hoje, pois o Ministério não consegue

ter uma fiscalização eficiente; e a questão da regulamentação dos serviços de radiodifusão (*Código Brasileiro de Telecomunicações - Lei 4.117/1962*), que especifica de acordo com o artigo 14: “*A mesma entidade ou as pessoas que integram o seu quadro societário e diretivo não poderão ser contempladas com mais de uma outorga do mesmo tipo de serviço de radiodifusão na mesma localidade*”. Contudo, toda essa questão é, na verdade, uma briga de interesses entre os grandes grupos de radiodifusores, já que alguns são a favor, pois aumentaria a possibilidade de publicidade e rendimento, enquanto outros são contra, de acordo com a visão desses, iriam perder espaço (audiência) para outras TVs que estariam transmitindo duas, três ou até quatro programações distintas na TV aberta.

### 3.15.1.3 A TV Interativa

A TV Interativa pode ser definida como “qualquer coisa que torne possível aos telespectadores dialogarem com as pessoas que fazem um canal de televisão, programa ou serviço. Um diálogo que leva os telespectadores para além da experiência passiva de assistir e os permita fazer escolhas ou tomar ações - mesmo que as ações sejam tão simples como preencher um cartão postal e enviá-lo pelo correio, ou desenhar uma imagem na tela da televisão.” (Mark Gawlinski, BBC)

Essa definição vem do fato da TV Interativa já existir anteriormente ao advento da TV Digital, com uso de interações por vias indiretas, como telefone, correio, internet, sms, etc. Um dos primeiros programas interativos no mundo foi o *Winky Dink and You*, da CBS criado em 1953, um programa infantil. (Fonte: Teixeira, 2009).



Figura 35 – Primeiro programa interativo no mundo (Fonte: Teixeira, 2009)

Para a interatividade nos moldes da TV Digital, ela trata da interação mediada por interfaces eletrônicas dentre os três níveis abaixo:

- ✓ **Reativo:** nesse nível, as opções e realimentações (feedbacks) são dirigidas pelo programa, havendo pouco controle do usuário sobre a estrutura do conteúdo.
- ✓ **Coativo:** apresentam-se aqui possibilidades de o usuário controlar a seqüência, o ritmo e o estilo do conteúdo;
- ✓ **Pró-ativo:** o usuário pode controlar tanto a estrutura quanto o conteúdo.

Dessa forma, podemos dividir a interatividade na TV Digital em três tipos:

- ✓ **Interatividade Local:** A emissora transmite os programas e serviços com opções adicionais e o usuário pode escolher entre uma programação e outra ou habilitar/desabilitar opções;
- ✓ **Interatividade com canal de Retorno:** o usuário pode "responder" aos aplicativos enviados pela emissora, simultaneamente ou não a um programa;
- ✓ **Interatividade Plena:** Acesso à internet para publicação de conteúdo pessoal.

Com isso, surge algumas definições nas televisões interativas:

- ✓ **TV Expandida (eTV):** a interatividade como extensão do conteúdo da programação: enquetes, reality shows, propaganda, jogos, esportes, etc.
- ✓ **Serviços Interativos:** a TV como um terminal de acesso: previsão do tempo, horóscopo, sistema de e-mail, sms, TV banking, internet TV, etc.
- ✓ **Infra-estrutura de navegação:** Objetos da interatividade, Menus, guias de programação, etc.

### 3.15.1.4 Codificação de áudio

A codificação é o processo de transformação de sinais externos em bits que representem tais sinais. Tal processo se dá, por exemplo, através de amostragem, e a informação obtida pode ser ainda compactada.

A compressão de áudio é justificada pela limitação de banda e capacidade de dispositivos de armazenamento. Ela explora a limitação da percepção auditiva humana. A faixa audível para o ouvido humano vai de 20Hz a 20kHz e é mais sensível na faixa de 2 kHz a 4kHz.

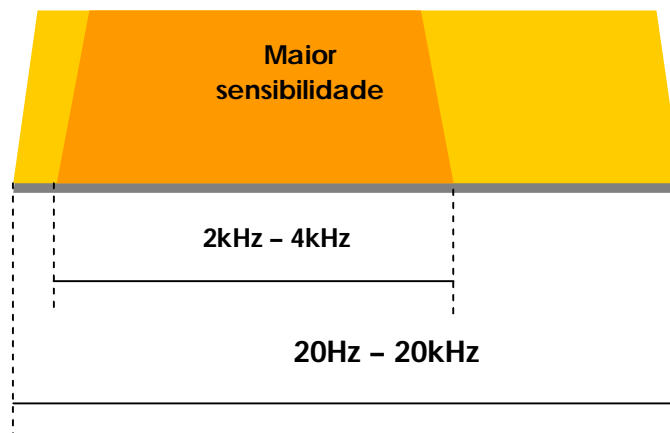


Figura 36 – Faixa audível – ouvido humano (Fonte: Cristóvam, 2010)

Os sinais de áudio devem obrigatoriamente ser codificados por uma combinação de codificação transformada no tempo e em frequência. A transformada em frequência deve decompor o sinal de entrada em seus componentes de frequência empregando a transformada discreta do cosseno modificada (MDCT – *Modified Discrete Cosine Transform*) quando a quantidade de informação é reduzida, diminuindo-se o desvio em frequência de cada componente. (Fonte: ABNT, 2007).

Como ferramenta de compressão adicional empregam-se ponderações psico-acústicas<sup>6</sup> do sinal, onde os códigos devem obrigatoriamente ser ponderados de forma a minimizar a degradação nas faixas de frequências perceptíveis pelo ouvido humano. A compressão de

<sup>6</sup> Esse termo vem da psicoacústica que estuda a percepção subjetiva das qualidades (características) do som: intensidade, tom e timbre. Estas qualidades ou características do som estão, por sua vez, determinadas pelos próprios parâmetros do som, principalmente, frequência e amplitude.

áudio e os procedimentos de transmissão devem obrigatoriamente ser compatíveis com a ISO/IEC 14496-3.

O decodificador deve obrigatoriamente ser construído assumindo-se que qualquer estrutura válida da ISO/IEC 13818-1, incluindo descritores privados, pode ser incluída no fluxo de bits mesmo que no início das transmissões esses descritores não sejam utilizados. O decodificador de áudio deve obrigatoriamente desconsiderar estruturas “reservadas” ou aquelas que correspondem a funções não implementadas pelo receptor. (Fonte: ABNT, 2008).

Abaixo, temos uma visão geral dos procedimentos de transmissão e codificação de áudio:

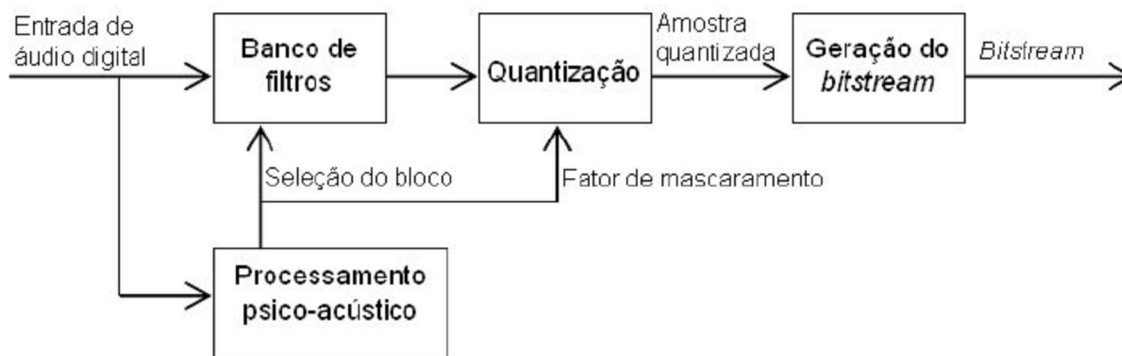


Figura 37 – Procedimentos de transmissão e codificação de áudio (Fonte: ABNT, 2008)

O banco de filtros deve obrigatoriamente converter a entrada de sinal de áudio-digital do domínio do tempo para o domínio da frequência. A partir de então, o banco de filtros aplica a transformada discreta dos cossenos modificada e funções de janelamento a blocos do sinal de entrada, de acordo com características fisiológicas audíveis.

Os processos psico-acústicos devem obrigatoriamente calcular a quantidade de mascaramento (limite para o discernimento entre um sinal específico de áudio de outros sinais) e alimentam o banco de filtros com blocos do sinal de entrada.

As amostras devem obrigatoriamente ser quantizadas após o processamento pelo banco de filtros, com base no fator de mascaramento calculado pelos processos de psico-acústica. Deste modo, o número total de bits utilizado por cada bloco não deve ser excedido.

Os formatos de compressão de áudio na TV Digital são:

- ✓ MPEG-4 AAC LC multicanal 5.1 e níveis inferiores;

- ✓ MPEG-4 HE-AAC estéreo.

Os seguintes perfis e níveis do padrão MPEG-4 AAC devem obrigatoriamente ser permitidos:

- ✓ LC (*low complexity*), perfil básico do padrão AAC; níveis L2 e L4;
- ✓ HE (*high efficiency*), perfil avançado de alta eficiência, combinando o perfil LC com o uso da ferramenta SBR (*spectral band replication*) para a versão 01 deste perfil, níveis L2 e L4;
- ✓ HE combinado à ferramenta PS (*parametric stereo*) para a versão 02 deste perfil; nível L2.

O perfil e o nível do codificador MPEG-4 AAC devem obrigatoriamente ser sinalizados conforme ABNT NBR 15602-3 e ABNT NBR 15603-2.

### 3.15.1.5 Codificação de vídeo

Os sinais digitais de vídeo nas resoluções SD e HD, à entrada do sistema de radiodifusão, devem obrigatoriamente ser compostos por um sinal representando a luminância da cena (sinal de luminância) e por dois outros sinais representando as características de croma (sinais de complemento de cor, pois estes resultam de operações de diferenças de cores). Os sinais de luminância e complemento de cor devem obrigatoriamente ser determinados pelas seguintes equações: (Fonte: ABNT, 2008)

$$Y = INT[219DEY' + 16D + 0,5]$$

$$CR = INT[224DEC' R + 128D + 0,5]$$

$$CB = INT[224DEC' B + 128D + 0,5]$$

Onde,

- ✓  $INT [A]$  representa a parte inteira de um número real  $A$ ;
- ✓  $Y$  é o valor numérico do sinal digital de luminância;



- ✓  $CR$  e  $CB$  são os valores numéricos dos sinais digitais de complemento de cor, em relação ao vermelho e ao azul, respectivamente;
- ✓  $D$  é o valor numérico substituído nas equações por “1” ou “4”, de acordo com a quantidade de bits usados na quantização, 08 ou 10 bits, respectivamente;
- ✓  $EY'$  ,  $EC'$  ,  $R$ ,  $EC'$  e  $B$  são os valores numéricos dos sinais analógicos da luminância e dos complementos de cor e devem obrigatoriamente respeitar as equações dadas na tabela abaixo:

Tabela 08 – Equações dos sinais analógicos e dos complementos de cor (Fonte: ABNT, 2008)

SD	HD
$E'_Y = 0,299E'_R + 0,587E'_G + 0,114E'_B$	$E'_Y = 0,2126E'_R + 0,7152E'_G + 0,0722E'_B$
$E'_{CR} = (E'_R - E'_Y) / 1,402$	$E'_{CR} = (E'_R - E'_Y) / 1,5748$
$E'_{CB} = (E'_B - E'_Y) / 1,772$	$E'_{CB} = (E'_B - E'_Y) / 1,8556$

O emprego da colorimetria HD da Tabela 08 em sinais SD também é permitido. Os valores de  $ER'$  ,  $EG'$  e  $EB'$  devem obrigatoriamente representar os níveis de tensão (normalizados em relação ao sinal branco de referência) resultantes da correção gama (realizada no lado do receptor para recuperar os sinais  $ER$  ,  $EG$  e  $EB$  , com características opostas às do *display*, de tal forma que se obtenha a correta reprodução da intensidade das componentes de cor vermelha, verde e azul na tela do televisor) dos níveis de tensão dos sinais vermelho, verde e azul, obtidos na reprodução de um pixel.

$ER'$  ,  $EG'$  e  $EB'$  devem obrigatoriamente ser aplicados a *displays* com os valores de "x" e "y" apresentados na tabela 09, onde "x" e "y" são as coordenadas para os valores quantitativos das cores vermelha, verde e azul no diagrama de cromaticidade definido na ISO/IEC 10527.

Tabela 09 – Valores das coordenadas X e Y para as componentes vermelha, verde e azul (Fonte: ABNT, 2008)

Componentes	SD		HD	
	x	y	x	y
Vermelho	0,67	0,33	0,640	0,330
Verde	0,21	0,71	0,300	0,600
Azul	0,14	0,08	0,150	0,060

Algoritmos de compressão de vídeo como o MPEG-4 e o H.264 usam a previsão entre quadros (*interframe prediction*) para reduzir os dados de vídeo entre uma série de quadros. Isso envolve técnicas tais como codificação de diferenças, onde um quadro é comparado com um quadro de referência, e apenas os pixels que se modificaram em relação ao quadro de referência são codificados. Dessa forma, reduz-se o número codificado e enviado de valores de pixels. Quando essa seqüência codificada é exibida, as imagens aparecem exatamente como na seqüência de vídeo original.

Abaixo vemos uma figura que exemplifica essa codificação:



Figura 38 – Codificação de diferenças – Frames I e Frames P (Fonte: Axis, 2008)

Com a codificação de diferenças, apenas a primeira imagem (quadro I) é integralmente codificada. Nas duas imagens seguintes (quadros P), são feitas referências aos elementos estáticos da primeira imagem, ou seja, a casa. Apenas as partes móveis, ou seja, o homem correndo, são codificadas através de vetores de movimento, reduzindo, assim, a quantidade de informação enviada e armazenada. Para maiores informações sobre outras técnicas, como a compensação de movimento por blocos, consultar ISO/IEC 14496-10.

Um quadro I, ou intraquadro, é um quadro autônomo que pode ser codificado de maneira independente, sem nenhuma referência a outras imagens. A primeira imagem de uma seqüência de vídeo sempre é um quadro I. Os quadros I são necessários como pontos de início

para novos visualizadores ou como pontos de resincronização, caso o fluxo de bits transmitido seja danificado. Os quadros I podem ser usados para implementar funções de avanço rápido, retrocesso e outras funções de acesso aleatório. Um codificador insere automaticamente quadros I a intervalos regulares ou sob demanda caso seja esperado que novos clientes entrem na visualização de um fluxo. A desvantagem dos quadros I é que eles consomem muito mais bits, mas, por outro lado, não geram muitos artefatos, que são causados por dados ausentes.

Um quadro P, que significa “interquadro preditivo” (predictive inter frame), faz referência a partes de quadros I e/ou P anteriores para codificar o quadro. Os quadros P normalmente requerem menos bits que os quadros I, mas uma das desvantagens é que eles são muito sensíveis a erros de transmissão devido à complexa dependência de quadros P e/ou I anteriores.

Um quadro B, ou interquadro bipreditivo (bi-predictive inter frame), é um quadro que faz referências a um quadro de referência anterior e a um quadro futuro. O uso de quadros B aumenta a latência.

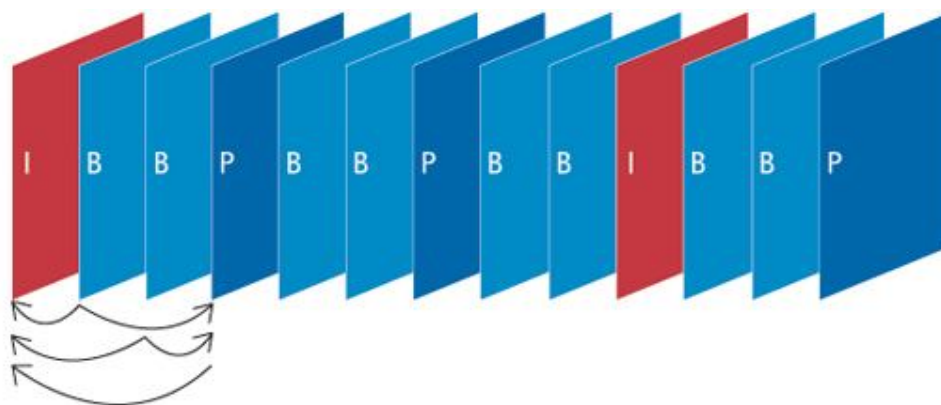


Figura 39 – Sequência com frames I, B e P (Fonte: Axis, 2008)

Um P-frame pode fazer referência apenas aos quadros I ou P precedentes, ao passo que um B-frame pode fazer referência a quadros I ou P precedentes e posteriores. Quando um decodificador de vídeo restaura um vídeo decodificando o fluxo de bits quadro a quadro, a decodificação deve sempre começar com um I-frame. Os Quadros P e B, se forem usados, devem ser decodificados junto com o(s) quadro(s) de referência.

Convém que o processo de estimação de movimento seja realizado para cada bloco resultante do particionamento do macrobloco. Esse particionamento pode gerar blocos de

tamanho 16 x 16, 16 x 8, 8 x 16, 8 x 8, 8 x 4, 4 x 8 ou 4 x 4 amostras de luminância, como mostrado na figura 41.

O processo de predição intra pode ser realizado para todo o macrobloco, com tamanho 16 x 16 amostras ou para cada um dos seus blocos 4 x 4 amostras. O processo deve obrigatoriamente ser realizado conforme a ITU-T Recommendation H.264 e conforme a ISO/IEC 14496-10:2005, subseção 8.1.

O processo de transformação e quantização deve obrigatoriamente ser realizado conforme a ITU-T Recommendation H.264 e conforme a ISO/IEC 14496-10:2005, subseção 8.5.

O filtro para remoção de efeito de blocos (*deblocking*) deve obrigatoriamente ser um processo condicional a ser aplicado nas bordas dos blocos de cada macrobloco, conforme a recomendação ITU-T Recommendation H.264 e conforme a ISO/IEC 14496-10:2005, subseção 8.7.

O esquema de codificação de vídeo usual está esquematicamente representado na figura abaixo:

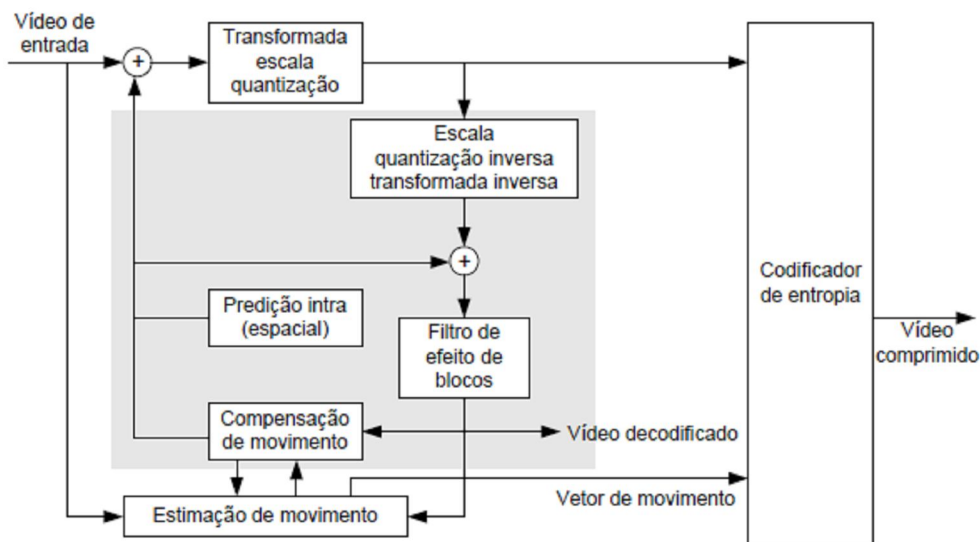


Figura 40 – Representação esquemática do processo de codificação (Fonte: ABNT, 2007)

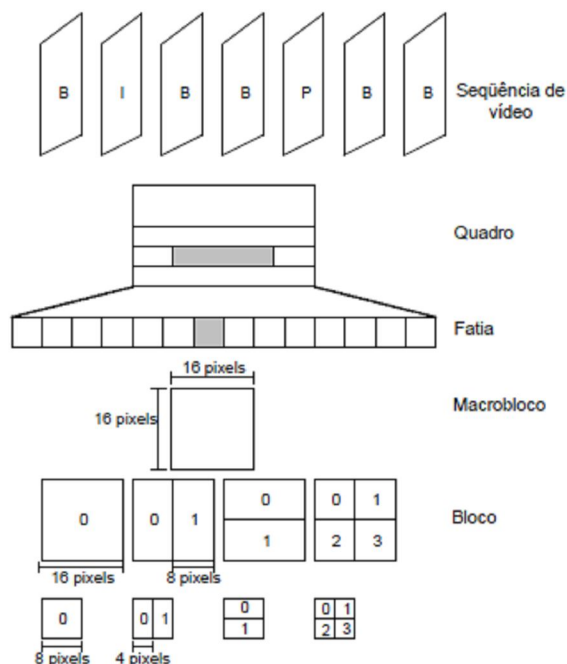


Figura 41 – Fracionamento da sequência de vídeo em macroblocos (Fonte: ABNT, 2007)

### 3.15.1.6 Transmissão

O Sistema Brasileiro de TV Digital foi projetado para prover uma proteção robusta contra interferências de onda de rádio a fim de proporcionar a transmissão de vídeo e som de alta qualidade para os receptores de TV. O sistema é composto pelos blocos funcionais mostrados na figura 43. O bloco SBTVD-01 trata da funcionalidade da codificação de canal e modulação baseado na referencia ARIB STD 31 V 1.6. O bloco SBTVD-02 trata da codificação de vídeo e áudio que se refere aos codificadores H264/AVC e MPEG-4/AAC respectivamente. O bloco SBTVD-03 descreve os procedimentos para multiplexar em um mesmo feixe de dados os diferentes transport stream enviados pelos codificadores. O bloco SBTVD-04 descreve as funcionalidades do terminal de acesso (Set Top Box). O documento SBTVD-05 trata da propriedade de direitos intelectuais. O bloco SBTVD- 06 descreve o uso do middleware e interatividade. Finalmente o SBTVD-07 padroniza o canal de retorno para aplicação em interatividade.

A estrutura de RF do transmissor é bastante semelhante ao sistema analógico, requerendo somente uma resposta de fase e linearidade de amplitude mais acurada para minimizar a distorção do sinal.

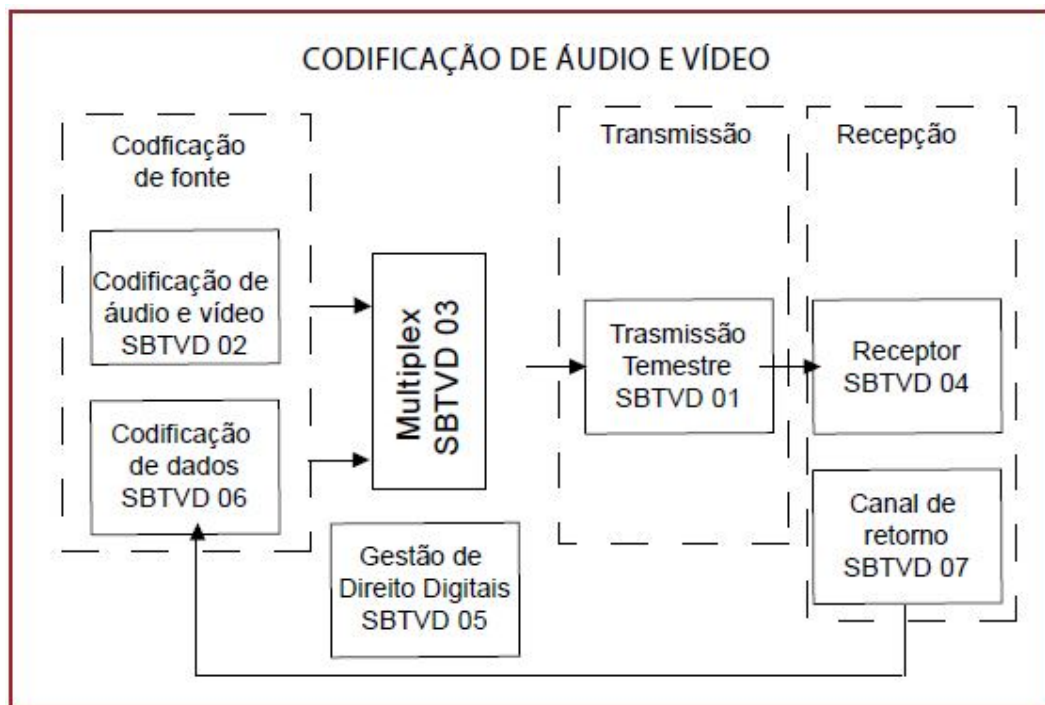


Figura 42 – Bloco funcional do sistema de transmissão do SBTVD (Fonte: Produção Profissional, 2008)

A transmissão BST-OFDM consiste na divisão da banda útil do canal em 13 segmentos de 428,5 kHz cada para formar até três distintos programas ou camadas. Os parâmetros de transmissão podem ser configurados individualmente para cada segmento, aqui referido como segmento OFDM, formando um canal de composição flexível. Este procedimento de configuração é designado estrutura de camada hierárquica.

Uma das características importantes da modulação OFDM é a possibilidade de operar no esquema de transmissão rede de frequência única (SFN). Para adequar a distância entre as estações SFN e robustez ao efeito Doppler durante a recepção móvel, foram estabelecidos três diferentes espaçamento entre as frequências portadoras, denominados modos. Esses espaçamentos são de 3.968 Hz para modo 01, 1984 Hz para modo 02 e 992 Hz para modo 03. Esses espaçamentos resultam em 108 portadoras para cada segmento OFDM no modo 1, 216 portadoras para o modo 02 e 432 portadoras para o modo 03.

No modo 01, cada segmento consiste de 108 portadoras, sendo 96 para transmissão de dados e 12 frequências piloto. No modo 02 esse número é duas vezes maior e no modo 3 é quatro vezes superior (vide tabela 10). Os dados apontados na figura 44 significam informações conectados ao middleware. A transmissão hierárquica consiste no envio

simultâneo de distintos programas A, B ou C ao mesmo tempo no mesmo canal e a cada um é permitido aplicar diferente configuração de codificação de canal. O segmento para recepção portátil está localizado no centro da fila de segmento como mostrado na figura 45.

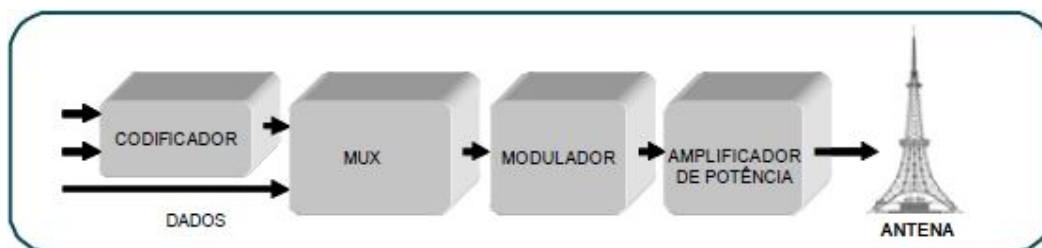


Figura 43 – Configuração do SBTVD (Fonte: Produção Profissional, 2008)

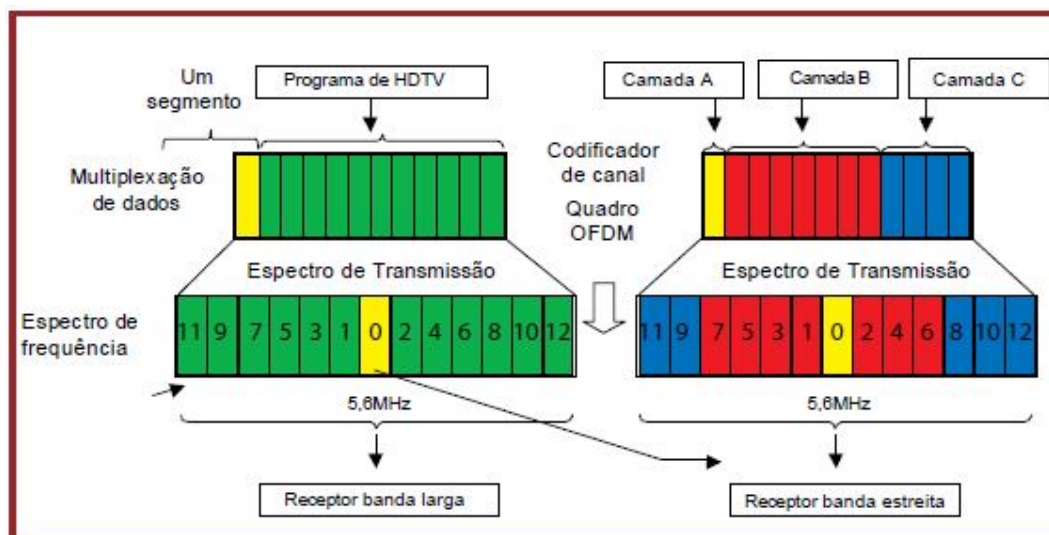


Figura 44 – Exemplo de configuração de camadas (Fonte: Produção Profissional, 2008)

Os receptores banda larga também são conhecidos como receptores fullseg e os receptores banda estreita como receptores one-seg. Os segmentos de ordem ímpar estão localizados no lado esquerdo e os de ordem par no lado direito em relação ao centro. Cada segmento pode ser configurado sem o envolvimento de outros segmentos. O número de segmentos agrupados em cada camada hierárquica pode ser selecionado pelo radiodifusor de acordo com a intenção de serviço que pretende oferecer. No sistema brasileiro é possível transmitir sinais de TV para receptor portátil de banda estreita. Esse método é denominado recepção parcial e usa sempre o segmento central da banda de 6MHz. Ele é considerado

também como uma camada hierárquica. A figura 45 apresenta dois exemplos de transmissão hierárquica. O exemplo da esquerda mostra a transmissão simultânea de um programa de TV em alta definição e um programa para receptor portátil. Na figura 45, o exemplo da direita mostra a transmissão simultânea de três programas em diferentes camadas hierárquicas.

O circuito de transmissão é dividido em três seções: Codificação de canal, modulador e seção de RF, com mostra a figura 46. O módulo multiplexador reúne até três feixes de transport stream (TS) proveniente de distintos codificadores para formar um único feixe para ser submetido ao bloco corretor de erros Reed Solomon (204, 188). Após este estágio o TS é dividido novamente nos seus conteúdos originais em pacotes de 204 bytes (TSP) para ser submetido ao codificador convolucional (codificação interna).

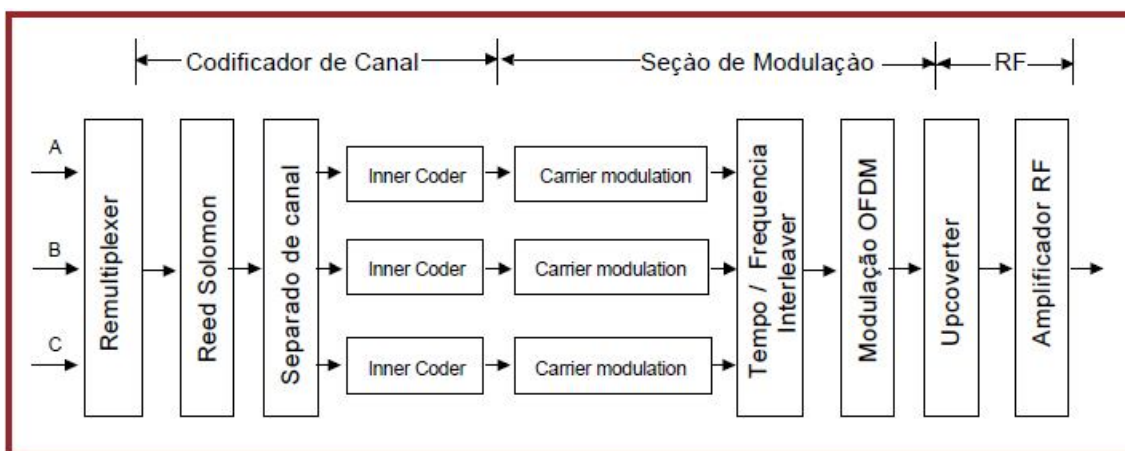


Figura 45 – Configuração básica do transmissor digital (Fonte: Produção Profissional, 2008)

Os principais valores do sistema brasileiro estão mostrados na tabela 10, abaixo (A taxa de bits de transmissão assume valores diferentes dependendo do esquema de modulação, taxa de código convolucional e intervalo de guarda.):



Tabela 10 - Resumo das características com os principais valores do SBTVD

Descrição		Valor do sistema		
Número de segmento OFDM		13		
Largura de banda do segmento		428,57 kHz		
Modo		1	2	3
Portadora por segmento		108	216	432
Portadora total por canal 6MHz		1405	2809	5617
Espaçamento entre portadoras		3.968Hz	1.984 Hz	992 Hz
Comprimento do símbolo		252 $\mu$ s	504 $\mu$ s	1.008 $\mu$ s
Intervalo de guarda $\mu$ s	1/4	63	126	252
	1/8	31,5	63	126
	1/16	15,75	31,5	63t
	1/32	7,87	15,75	31,5t
Comprimento do símbolo por segmento $\mu$ s	1/4	315	628	1260
	1/8	283,5	565	1134
	1/16	267,7	533,5	1071
	1/32	259,8	517,7	1039,5
Tipo de modulação		QPSK, 16QAM, 64QAM		
Símbolo por quadro		204		
Codificação interna		Codificação convolucional $\frac{1}{2}$ , $\frac{2}{3}$ , $\frac{3}{4}$ , $\frac{5}{6}$ , $\frac{7}{8}$		

### 3.15.1.6.1. Seção de Codificação de Canal

O esquema de codificação de canal tem como objetivo introduzir alguns algoritmos ao sinal para auxiliar o receptor a reconhecer e corrigir os erros causados pelo canal de transmissão. A figura 47 mostra os estágios de processamento de bits:

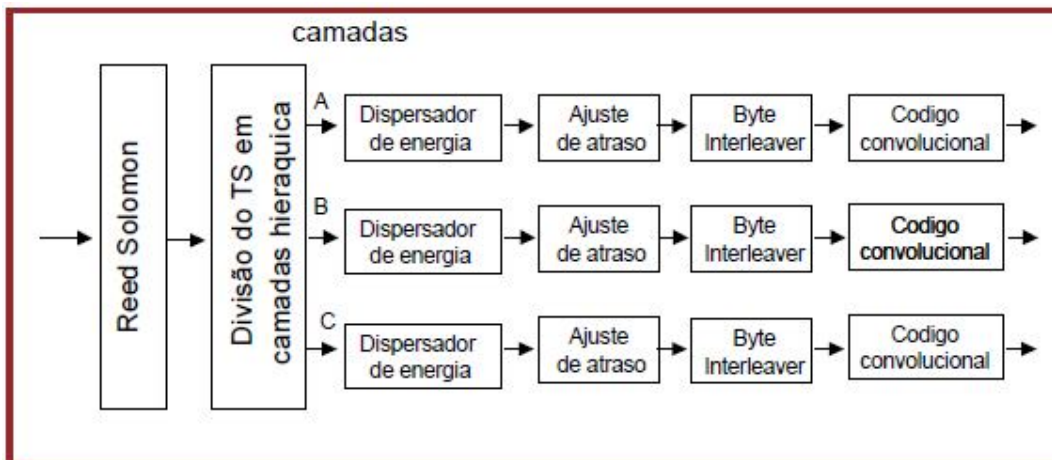


Figura 46 – Seção de codificação de canal (Fonte: Produção Profissional, 2008)

O Reed Solomon é um código corretor de erros em blocos que é aplicado coletivamente para o transport stream total que irá formar o pacote de dados do canal. Em cada símbolo de 188 bytes é adicionado mais 16 bytes de paridade o qual é capaz de corrigir até 8 bytes errados. No caso da transmissão hierárquica o transporte stream resultante é novamente dividido no conjunto de informações dos pacotes originais, num máximo de três stream paralelos de processamento. A seguir o dispositivo dispersor de energia, cujo objetivo é evitar a repetição de grande seqüência de 1 ou 0, é aplicado em cada seção de processador paralelo usando um circuito PRBS (Pseudo Random Bit Sequence). O ajuste de atraso associado ao bit interleaving objetiva a compensação de tempo para equalizar o tempo de transmissão e recepção de todas as camadas e é sempre conduzido pelo lado da transmissão.

A soma de todos os atrasos incluindo o atraso de transmissão e recepção causado pelo bit interleaving é sempre equivalente ao comprimento de um quadro. O codificador interno é um convolucional puncionado com código mãe de  $\frac{1}{2}$  e tendo o comprimento de compressão  $k$  de 7. A seguir é efetuado o puncionamento para a taxa de  $\frac{1}{2}$ ,  $\frac{2}{3}$ ,  $\frac{3}{4}$ ,  $\frac{5}{6}$  e  $\frac{7}{8}$ .

Robustez e flexibilidade podem ser conseguidos especificando diferentes conjunto de parâmetros de transmissão tais como o número de segmento, taxa de codificação interno e esquema de modulação para diferentes camadas hierárquicas. (Fonte: Produção Profissional, 2008)

### 3.15.1.6.2. Modulação de bits

Esta seção descreve a modulação da seqüência de bits fornecidos pela seção de codificação de canal. No processo de modulação das portadoras os bits do sinal de entrada são entrelaçados e mapeados pelo esquema definido para cada camada hierárquica.

O sinal de entrada deve ser de 2 bits por símbolo para modulação em QPSK, mapeado para os eixos I e Q, de 4 bits por símbolo para modulação em 16QAM mapeados para os eixos I e Q e de 6 bits por símbolo para modulação e 64QAM mapeados para os eixos I e Q. Como o número de bits por símbolo aumenta de 2 para 4 e para 6, a taxa de bits aumenta na mesma proporção. Ao mesmo tempo a distância entre portadoras também diminui e a configuração fica menos robusta.

Para proceder ao mapeamento são inseridos na entrada do mapeador atraso de 120 elementos de bits no momento do entrelaçamento de bits para a modulação em QPSK. Para proceder ao mapeamento em 16QAM não é introduzido atraso no primeiro bit, introduzido atraso de 40 elementos de bits para o segundo bit, atraso de 80 elementos de bits para o terceiro bit e 120 elementos de bit para o quarto bit. Veja na figura abaixo:

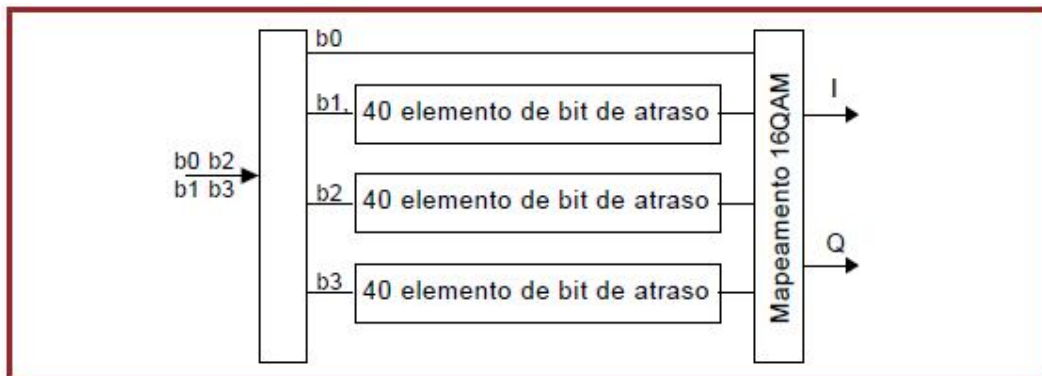


Figura 47 – Diagrama de modulação 16-QAM (Fonte: Produção Profissional, 2008)

Observe que existe uma correlação entre a taxa de bit transmitido e a robustez do sinal contra os efeitos da interferência. Então, considerando um intervalo de guarda de 1/8 na modulação QPSK com taxa de C/N de 10dB há uma recepção com excelente qualidade, entretanto a taxa de bits transmitida é de 10Mbps. Para a modulação em 64QAM necessita-se

de C/N de 18 dB para garantir uma boa recepção, contudo a taxa de bits transmitido é de 20Mbps aproximadamente.

Devido ao fato do nível de energia das portadoras modulado com alto número de estados ser maior que aquele modulado com pequeno número de estados, o nível do sinal de transmissão precisa ser normalizado para que as potências médias das portadoras fiquem aproximadamente constante, independente do esquema de modulação utilizado. A tabela 11 mostra os fatores propostos:

Tabela 11 – Fatores propostos de modulações (Fonte: Produção Profissional, 2008)

Modulação	$Z /$
QPSK	$Z / \sqrt{2}$
16QAM	$Z / \sqrt{10}$
64QAM	$Z / \sqrt{42}$

A seguir os sinais de diferentes camadas hierárquicas parametrados para diferentes configurações necessitam ser combinados para serem submetidos em comum ao processo de IFFT (Inverse Fast Fourier Transform). Os sinais assim processados são submetidos a time interleaving em unidade de símbolos de modulação, para assegurar melhor robustez contra interferência de fading e também passa por processo de frequency interleaving, ação essa que reforça o efeito do time interleaving. A figura 49 ilustra essa configuração.

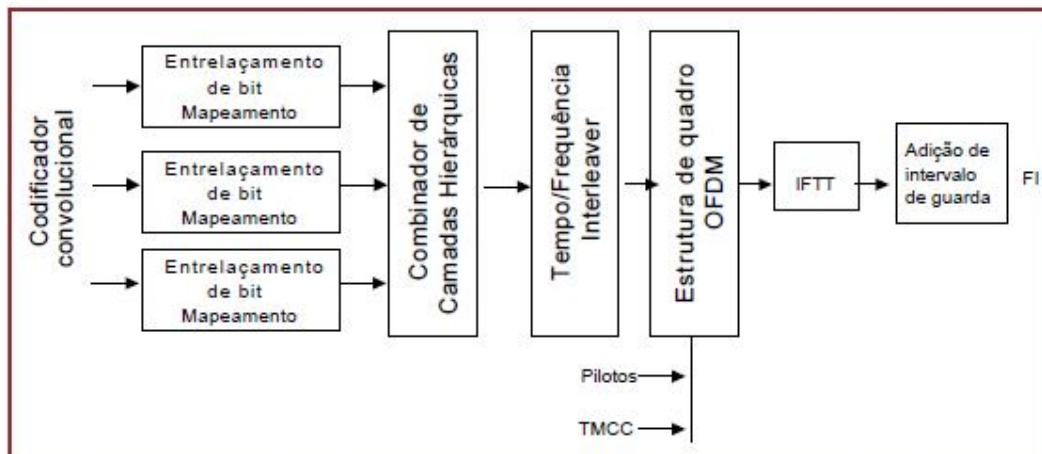


Figura 48 – Configuração para modulação das portadoras (Fonte: Produção Profissional, 2008)

Na estrutura de quadro são adicionados os seguintes sinais de piloto:

- ✓ TMCC: Sinal que conduz as informações de controle. O TMCC suporta o receptor na demodulação e decodificação de várias informações incluindo identificação de parâmetros de transmissão, indicador de chaveamento, flag para alarme de emergência, informação de configuração hierárquica atual e parâmetros para a próxima comutação. O piloto é transmitido em BQPSK provendo extremamente robusta informação de controle como o código de sincronismo;

- ✓ CP: Piloto contínuo serve como sinal de referência para a sincronização e informação para estimação e equalização de canal;

- ✓ SP: Piloto espalhado é inserido no segmento a cada 12 portadoras de dados, dentro de cada fila na direção do quadro OFDM e cada 4 símbolos na direção do símbolo (colunas). Ele representa 8% da energia total transmitida;

- ✓ AC: Piloto auxiliar é um sinal de extensão conduzindo informação adicional para controle do sinal de modulação.

O sinal emergente da estrutura do quadro OFDM é submetido ao IFFT que gera o sinal de FI de 44MHz. Como o sinal OFDM é constituído por diversas portadoras ortogonalmente modulado, cada símbolo é considerado como um elemento de comprimento  $TU$ .

Após a modulação OFDM, é inserido ao sinal o intervalo de guarda que é uma extensão cíclica do símbolo OFDM. Após a FI (Adição de intervalo de guarda), o sinal é convertido para a frequência do canal e submetido ao amplificador de potência.

### 3.15.1.6.3. Multiplexador (MUX)

Responsável por receber todos os sinais provenientes dos codificadores de áudio, vídeo e dados e concatená-los em um único fluxo de dados (Transport Stream - TS). Insere também as tabelas PSI/SI para permitir que o receptor consiga identificar corretamente os diferentes fluxos e se “auto configurar”.



Figura 49 – Fluxo de dados ou Transport Stream (TS) (Fonte: SET, 2009)

O Sistema de multiplexação é composto por dois tipos informações:

- ✓ As informações **PSI (Program Specific information)**: Padronizada pelo MPEG-2 System.
- ✓ As Informações **SI (Service Information)**: São características de cada sistema de transmissão (ISDB-T, DVB, ATSC e SBTVD).

O TS é originado por vários PES (Packetized Elementary Stream). Estes PES possuem informação de controle de campos, como também para transporte de dados. Posteriormente esses PES são subdivididos em pacotes menores e cada programa de tv vai ser constituído por vários destes pacotes. Por esta razão, o fluxo de dados vai incluir várias bases de tempo, possível graças ao fato do TS ter um mecanismo que permite a transmissão de múltiplos sinais de relógio PCR (Program Clock Reference). Cada programa é mais tarde regenerado no decodificador graças ao PCR.

As tabelas PSI/SI são tabelas responsáveis pela identificação dos serviços que trafegam no transporte stream, como pode ser visto na tabela 12.

Tabela 12 – Tabelas PSI/SI

<b>PAT</b>	Tabela de Associação de Programas	<b>PSI</b>
<b>PMT</b>	Tabela de Mapeamento de Programas	
<b>NIT</b>	Tabela de Informações de Rede	
<b>SDT</b>	Tabela de Descrição de Serviços	<b>SI</b>
<b>TOT</b>	Tabela de Diferença de Data e Fuso Horário	
<b>EIT</b>	Tabela de Informação de Eventos	
<b>BIT</b>	Tabela de Informação do Radiodifusor	
<b>ST</b>	Tabela de Preenchimento (invalida outras tabelas)	

Abaixo, vemos as principais características de cada uma delas:

✓ Tabelas PSI:

→ PAT – *Program Association table*

- Mandatória;
- Responsável por identificar as PMTs e a NIT;
- Igual a especificada pelo MPEG-2 System.

→ NIT – *Network information Table*

- Mandatória;
- A NIT do SBTVD é diferente da especificada pelo MPEG-System;
- Responsável por informar as configurações da rede e canal virtual.

→ PMT – *Program Map table*

- Mandatória;
- A PMT deve conter as informações que permitam o receptor localizar corretamente cada um dos conteúdos enviados, sejam eles vídeo, áudio ou dados.

→ CAT – *Conditional Access Table*

- Mandatória no caso de se utilizar acesso condicional

✓ Tabelas SI:

→ BAT: Bouquet Association Table

→ SDT: Service Description Table – **Mandatória**

→ EIT: Event Information Table – **Mandatória se EPG**

→ RST: Running Status Table

→ TDT: Time Date Table

→ TOT: Time Offset Table – **Mandatória**

→ ST: Stuffing Table

→ PCAT: Partial Content Announcement

→ BIT: Broadcaster Information – **Mandatória**

→ LIT: Local Event Information

→ ERT: Event Relation Table

→ ITT: Index Transmission Table

✓ Principais tabelas SI:

→ EIT: Para que as informações referentes ao conteúdo sejam enviada de acordo com o programa que está sendo transmitido, utiliza-se a tabela EIT.

→ SDT: Carrega as informações referentes aos serviços disponíveis no TS:

- Nome de cada serviço;
- Tipo de cada serviço (Fixo ou dados).



→ BIT: Responsável pelas informações de identificação da emissora, como:

- Nome da emissora, rede a qual ela pertence.

→ TOT: Carrega as informações de data e horário. Permite enviar uma informação que configura automaticamente o STB, no caso de mudança de horário. ex: Horário de verão.

#### 3.15.1.6.4. Modulações em DVB-S2

Os conceitos fundamentais para a concepção do DVB-S2 foram o aumento do desempenho na transmissão e mais flexibilidade sem aumentar demasiadamente a complexidade do receptor. Nesse sentido, o DVB-S2 aproveita os desenvolvimentos para melhorar os campos da codificação de canal e modulação, alcançando um ganho de 30% em capacidade em comparação ao DVB-S, adota um sistema de modulação e codificação adaptativa (AMC), que otimiza a codificação e modulação utilizadas para atingir a melhor eficiência. As modulações permitidas pela norma para a transmissão são QPSK, 8PSK, 16APSK, e 32 APSK. (Fonte: IT, Image Group)

Na modulação APSK (Modulação por Variação de Amplitude e Deslocamento de Fase), a amplitude e a fase da portadora são moduladas ao mesmo tempo e, conseqüentemente, muda simultaneamente as características de amplitude e a fase dos símbolos portadores para compor cada possível seqüência de bits em uma palavra binária.

A figura 50 ilustra as modulações para DVB-S2.

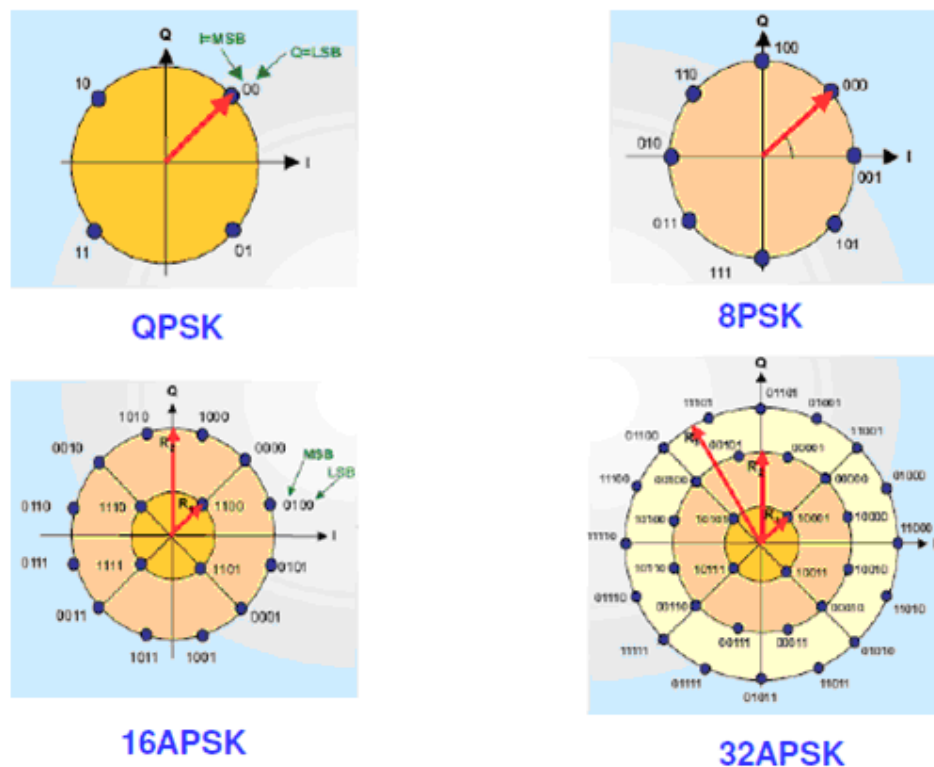


Figura 50 – Modulação DVB-S2 (Fonte: SET, 2009)

### 3.15.1.7 Carrossel de objetos

O carrossel de dados é um mecanismo do transporte que permite que um servidor de aplicações de um sistema de difusão de dados apresente, de forma cíclica, um conjunto de dados a um decodificador, repetindo o conteúdo do carrossel uma ou mais vezes. Se uma aplicação necessitar de um dado particular do carrossel dos dados, o decodificador deverá esperar a próxima vez que os dados pedidos forem transmitidos.

O nome “carrossel” vem do comportamento cíclico de envio dos dados que gera um *stream* elementar de dados que segue o protocolo *Digital Storage Media Command and Control* (DSM-CC), sub-protocolo *DSMCC Object*, do padrão MPEG-2. O DSM-CC foi desenvolvido para oferecer diversos tipos de serviços multimídia, entre eles a transmissão de dados multiplexados com o conteúdo audiovisual em um fluxo de transporte.

O padrão DSM-CC especifica três tipos de objetos: arquivos, diretórios e eventos. Assim, o carrossel de objetos pode possuir um verdadeiro sistema de arquivos, isto é, um

conjunto de diretórios e arquivos que, por exemplo, formam uma aplicação a ser executada nos terminais de acesso.

As especificações DSM-CC determinam que os dados transmitidos através do carrossel de objetos devem ser divididos em unidades denominadas módulos. Cada módulo pode possuir mais de um arquivo desde que não ultrapasse um total de 64 Kbytes. Os arquivos que estão em um mesmo módulo podem fazer parte de diretórios diferentes. Um arquivo que possui mais de 64 Kbytes deve ser transmitido em um único módulo, pois não é permitido dividir um arquivo em mais de um módulo.

Uma vez que os objetos foram dispostos em módulos, cada módulo é então transmitido, um após o outro. Após transmitir o último módulo, a transmissão é reiniciada desde o início. O resultado disso é um fluxo elementar que contém o sistema de arquivos transmitido de forma cíclica. Assim, se um determinado terminal de acesso não recebeu uma parte de um módulo em particular (devido a um erro na transmissão ou por ter sido iniciado após a transmissão desse módulo), basta esperar pela retransmissão desse módulo.

Em alguns casos, utilizar o carrossel de objetos dessa forma significa introduzir retardos impraticáveis para os dados enviados pelo provedor de conteúdo. Por exemplo, para um carrossel que possui tamanho total de 172 Kbytes transmitidos a uma taxa de 128 Kbps, são necessários aproximadamente 11 segundos para a transmissão de um ciclo completo. Assim, nesse exemplo, existe um retardo máximo de 11 segundos para carregar um arquivo qualquer que faz parte desse carrossel. Para amenizar esse problema, os geradores de carrossel oferecem como opção transmitir alguns módulos com maior frequência que outros. Assim, os módulos que contêm arquivos com maior prioridade podem ser transmitidos com maior frequência.

A figura 51 traz uma representação do carrossel de objetos.

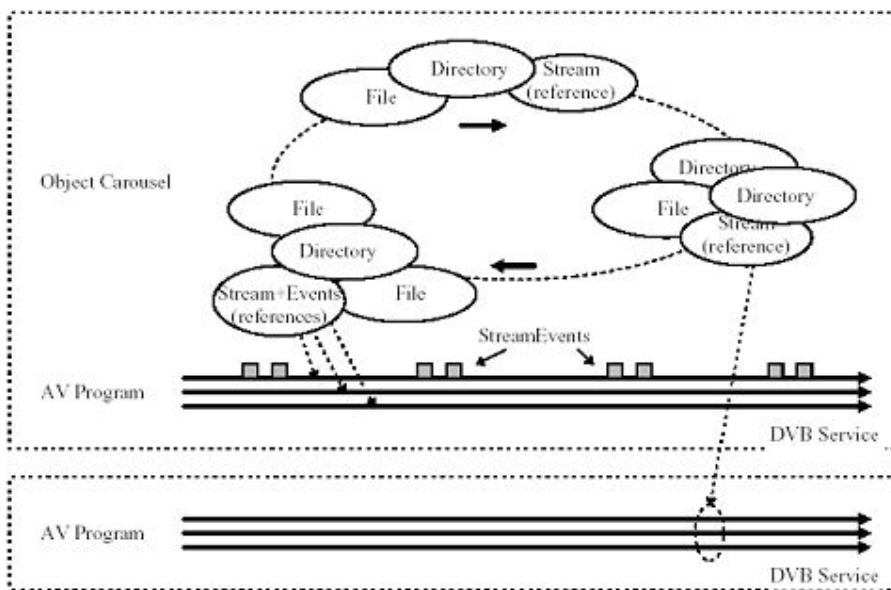


Figura 51 – Representação do carrossel

### 3.16 Middleware Ginga

Middleware é uma camada de software posicionada entre o código das aplicações e a infra-estrutura de execução, como ilustrado na figura 52.

Um middleware para aplicações de TV digital consiste de máquinas de execução das linguagens oferecidas, e bibliotecas de funções, que permitem o desenvolvimento rápido e fácil de aplicações.

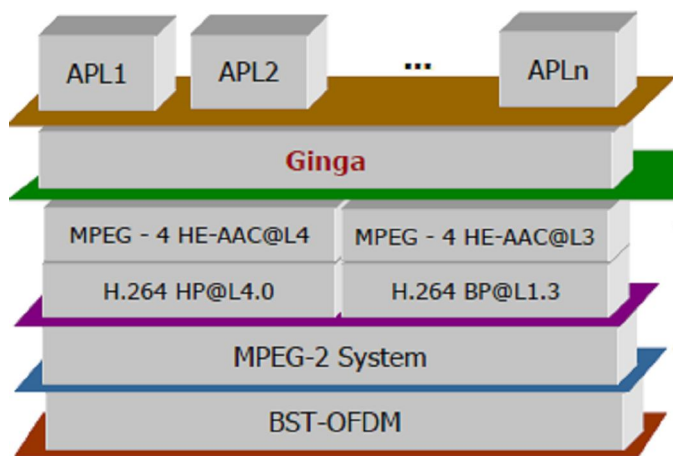


Figura 52 – Camada Middleware Ginga (Fonte: Soares e Barbosa, 2009)

O universo das aplicações de TV Digital pode ser particionado em um conjunto de aplicações declarativas e um conjunto de aplicações imperativas.

A entidade inicial de uma aplicação, isto é, aquela que dispara a aplicação, é que define a que conjunto a aplicação pertence, dependendo se essa entidade é codificada segundo uma linguagem declarativa ou imperativa.

O nome “Ginga” é uma qualidade de movimento e atitude que os brasileiros possuem e que é evidente em tudo o que fazem. A forma como caminham, falam, dançam e se relacionam com tudo em suas vidas. Ginga é flexibilidade, é adaptação, qualidades inerentes ao middleware brasileiro.

### 3.16.1 Arquitetura Ginga

Como já foi mencionado no capítulo anterior, a arquitetura do Ginga pode ser dividida em três módulos principais: Ginga-CC, Ginga-NCL e Ginga-J, como mostra a figura 53. Os dois últimos módulos compõem a camada de Serviços Específicos do Ginga.

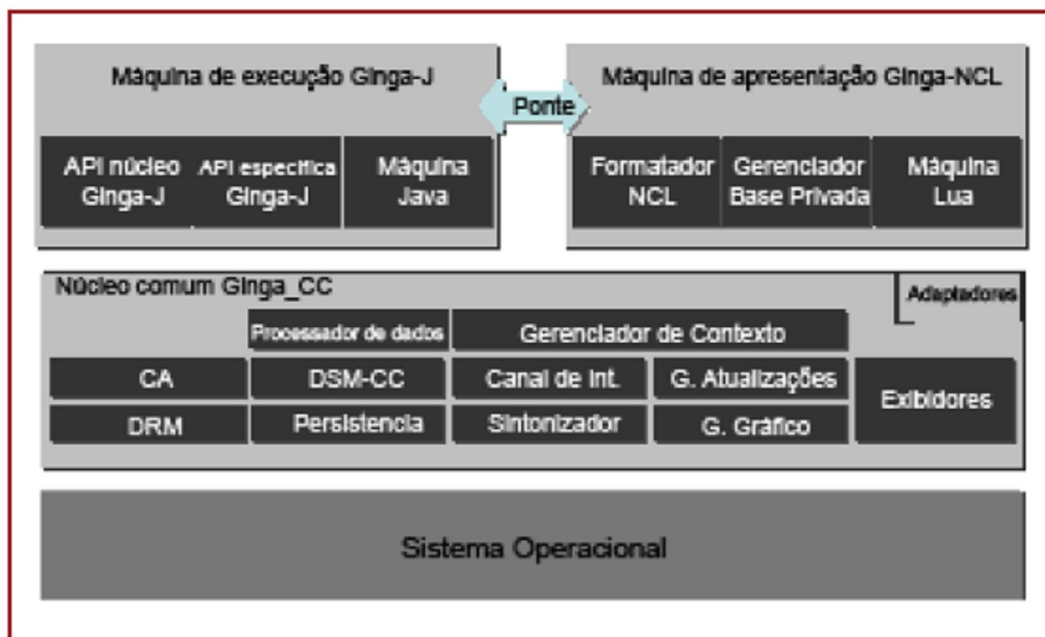


Figura 53 – Arquitetura Ginga (Fonte: Produção Profissional, 2008)

A figura 55 exemplifica a estrutura do ambiente de aplicações, onde é possível notar que há uma separação entre o núcleo destinado a aplicações NCL e a aplicações JAVA, e que entre eles há a ponte. A ponte é um mecanismo para aplicações que permite o mapeamento bidirecional entre as API JAVA e os objetos e métodos do DOM, ECMAScript e LUA Script, de acordo com a ABNT e o ambiente procedural, GINGA-J.

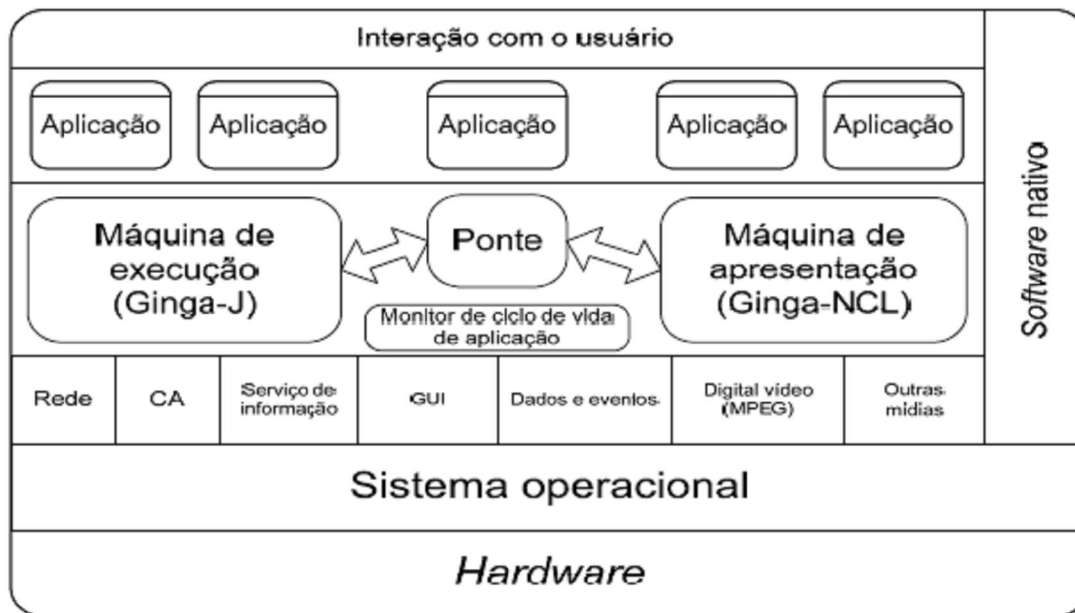


Figura 54 – Estrutura do ambiente de aplicações (Fonte: Produção Profissional/2008)

### 3.16.1.1 Ginga-NCL

O Ginga-NCL é a inovação totalmente brasileira do SBTVD. O ambiente tem por base a linguagem NCL (uma aplicação XML) e sua linguagem de script Lua, ambas desenvolvidas nos laboratórios da Pontifícia Universidade Católica do Rio de Janeiro.

Os ambientes declarativos dos sistemas americano, europeu e japonês têm por base a linguagem XHTML. Essa linguagem carrega o legado de tecnologias anteriormente desenvolvidas para navegação textual. Em sentido contrário, aplicações para TV Digital são usualmente centradas no vídeo. Além disso, o modelo da linguagem XHTML tem o foco no suporte à interação do usuário telespectador. Outros tipos de relacionamentos, como relacionamentos de sincronização espaço-temporal e relacionamentos para definição de

alternativas (adaptação de conteúdo e de apresentação), são usualmente definidos através de uma linguagem imperativa, no caso de todos os três sistemas citados a linguagem ECMAScript. (Fonte: Telemidia,2009)

Diferente das linguagens baseadas em XHTML, NCL define uma separação bem demarcada entre o conteúdo e a estrutura de uma aplicação, provendo um controle não invasivo da ligação entre o conteúdo e sua apresentação e leiaute. O modelo da linguagem NCL visa um domínio de aplicações mais amplo do que o oferecido pela linguagem XHTML. NCL visa não apenas o suporte declarativo à interação do usuário, mas o sincronismo espacial e temporal em sua forma mais ampla, tratando a interação do usuário como um caso particular. NCL visa também o suporte declarativo a adaptações de conteúdo e de formas de apresentação de conteúdo, o suporte declarativo a múltiplos dispositivos de exibição e a edição/produção da aplicação em tempo de exibição, ou seja, ao vivo.

### 3.16.1.2 Ginga-J

Como já mencionado, o ambiente imperativo Ginga-J oferece suporte a aplicações desenvolvidas usando a linguagem Java.

Ginga-J é dividido em três módulos: a máquina virtual Java; o núcleo e suas APIs, também chamadas APIs verde do Ginga-J; e o módulo responsável pelo suporte às APIs específicas do Ginga-J, chamadas de APIs amarela e vermelha do Ginga-J. Ginga-J segue às especificações da Norma ABNT NBR 15606-4:2010. As APIs verde do núcleo são as responsáveis por manter o sistema compatível o máximo possível com os sistemas americano e europeu, como ilustra a figura 55.

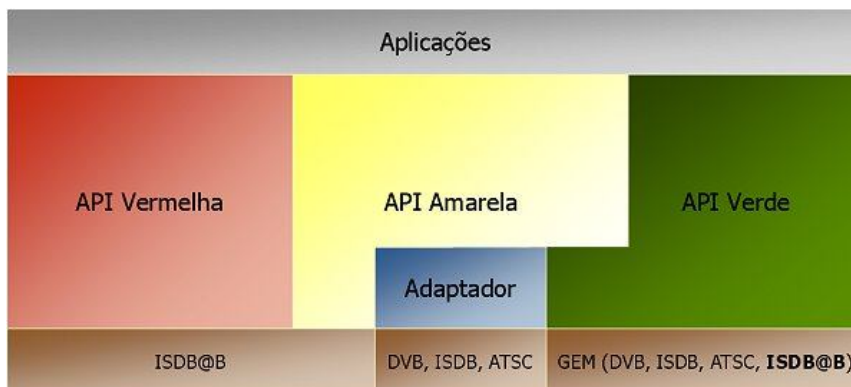


Figura 55 – APIs Ginga-J (Fonte: GingaDF, 2008)

As APIs específicas do Ginga que podem ser exportadas para outros sistemas são chamadas de amarelas. Entre elas estão aquelas que provêem suporte a múltiplos usuários, a múltiplos dispositivos e a múltiplas redes. Estão também aquelas que oferecem suporte às aplicações que podem ser recebidas, armazenadas e executadas em um tempo futuro. O suporte para as necessidades específicas de aplicações voltadas para o Brasil, em especial aplicações de inclusão social, são endereçadas pela API vermelha do Ginga-J. Ginga-J tem por base um conjunto de pacotes Java, comuns a diversos middlewares imperativos.

Entre as APIs específicas, cabe ainda ressaltar aquelas para a comunicação com o ambiente declarativo Ginga-NCL. Uma aplicação Java pode agir como entidade filha de uma aplicação declarativa, ou como uma entidade inicial controlando o ciclo de vida de uma entidade filha declarativa. Quando a entidade Java é a entidade inicial, ela pode criar, modificar e destruir documentos declarativos NCL através das APIs de comandos de edição Ginga, conforme especificado na Norma ABNT NBR 15606-2. Quando a entidade Java é uma entidade filha, ela atua como um objeto de mídia NCL, podendo se registrar para receber eventos NCL. Eventos NCL poderão, a partir de então, acionar métodos das classes Java do objeto. Objetos de mídia NCL com código imperativo Java podem também comandar condições de disparos de relacionamentos NCL, usados no sincronismo temporal e espacial da apresentação de conteúdos. Podem também manipular variáveis globais de aplicações declarativas, responsáveis pela determinação da adaptação de conteúdos ou da forma como conteúdos são apresentados.

As APIs amarela e vermelha, e as APIs da ponte são inovações brasileiras do ambiente imperativo Ginga-J, que o distingue dos demais ambientes imperativos dos middlewares do sistema europeu e americano.

### 3.16.1.3 Ginga-CC

Ginga-CC (Ginga Common Core) é o subsistema lógico que provê todas as funcionalidades comuns ao suporte dos ambientes declarativo, Ginga-NCL, e imperativo, Ginga-J. A arquitetura do sistema garante que apenas o módulo Ginga-CC deva ser adaptado à plataforma onde o Ginga será embarcado.



Ginga-CC provê, assim, um nível de abstração da plataforma de hardware e sistema operacional, acessível através de APIs bem definidas. Um conjunto de exibidores monomídia comuns faz parte dos componentes do Ginga-CC. As características de tais exibidores são definidas na Norma ABNT NBR 15606-1. Eles são exibidores de áudio, vídeo, texto e imagem, incluindo entre eles o exibidor MPEG-4/H.264, implementado por hardware. O acesso a tais exibidores se dá através de adaptadores, responsáveis por notificar eventos de apresentação e seleção (interação do usuário). Entre os exibidores também se encontra o exibidor (agente do usuário) HTML, especificado nas Normas ABNT NBR 15606-2 e ABNT NBR 15606-5. Na figura 54, o Gerenciador Gráfico é o responsável pelo gerenciamento do modelo conceitual do plano gráfico de apresentação. É ele que define o plano de exibição do vídeo principal H.264, os planos de exibição dos outros objetos de mídia que compõem uma aplicação TV Digital, e como esses planos se superpõem. A Norma ABNT NBR 15606-1 é responsável também por tal definição.

Todo o acesso a dados obtidos através do canal de retorno (ou canal de interatividade) é também de responsabilidade do Ginga-CC. No projeto proposto não haverá demonstração em banca com set-top-box, dessa forma, não serão apresentados maiores detalhes sobre o Ginga-CC. A simulação ocorrerá entre duas máquinas interligadas com o uso do Ginga-NCL set-top-box virtual.

### **3.17 Segurança**

O modelo de segurança Ginga é totalmente compatível com o modelo de segurança SBTVD. Ele lida com as mesmas áreas de segurança, ou seja, autenticação de aplicativos de difusão, políticas de segurança para aplicativos, segurança sobre o canal de interação e gerenciamento de certificados.

A autenticação de aplicativos Ginga-NCL deve obrigatoriamente ser realizada do mesmo modo para aplicativos Ginga-J. Se estiver assinado, o aplicativo deve obrigatoriamente seguir a estrutura de assinatura como especificado para o Ginga-J. Aplicativos Ginga-NCL não-autenticados irão operar dentro de um ambiente de caixa de areia (sand Box). Os aplicativos Ginga-NCL autenticados associados a um arquivo de solicitação de permissão podem ter permissões outorgadas fora da caixa de areia.

Outra questão que deve ser observada é a segurança da aplicação como o código fonte. NCL e Lua são linguagens que, teoricamente, não precisam ser compiladas para depois serem executadas. Em princípio, tanto o formatador NCL quanto o interpretador Lua executam ações que foram escritas num arquivo de texto comum, que pode ser lido e entendido por qualquer pessoa. Por isso, no desenvolvimento de aplicações há a necessidade de preocupação com a questão do acesso ao código-fonte da aplicação. Por mais que os conteúdos multimídia (áudio, vídeo, imagens e textos) de uma aplicação interativa já sejam protegidos por leis de direitos autorais, é necessário avaliar a proteção do código de uma aplicação. Uma das alternativas é o uso de compiladores que traduzam programas escritos na linguagem Lua para arquivos binários que podem ser carregados ou executados, como o Luac, que pré-compila um arquivo Lua, em ambiente GNU/Linux. (Fonte: ABNT, 2007)

### **3.18 O Canal de Retorno**

Como já foi dito no capítulo anterior, o canal de retorno faz parte do canal de interatividade, que é constituído pela interconexão das redes de televisão com as redes de telecomunicações, resultando nos dois caminhos de comunicação: Canal de Descida e Canal de Retorno.

A arquitetura recomendada para o sistema brasileiro de televisão digital é baseada em redes TCP/IP, possuindo servidores em qualquer localidade com acesso à internet, como mostra a figura 56. As diversas modalidades de redes de acesso permitem a conexão com a rede internet.

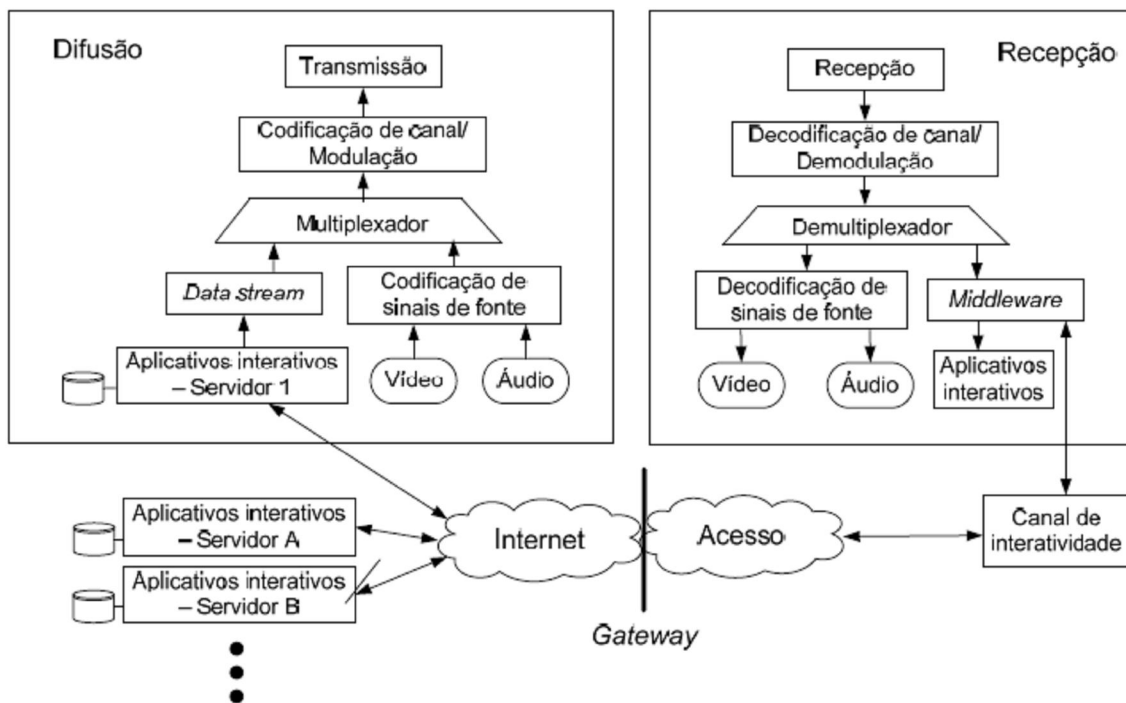


Figura 56 – Canal de interatividade bidirecional com acesso à internet (Fonte: ABNT, 2008)

As emissoras de televisão brasileiras defendem o uso da banda de TV VHF atual como canal de retorno. E como meio de acesso do canal de retorno nessa banda de 700 MHz seria utilizada a tecnologia sem fio denominada WiMAX, sendo portanto uma outra opção a ser acrescida à lista de meios de acesso convencionais (ADSL Internet, Cable Internet, GSM EDGE, GSM 3G, WiFi ou acesso discado). Essa idéia foi apresentada pelo Governo Brasileiro no Fórum WiMAX em Junho/2009 de forma a tentar criar um padrão internacional para o canal de retorno.

A internet será essencial nesse processo de canal de retorno dentro desse serviço de interatividade. Não temos ainda uma definição mais concreta quanto a isso, mas a banda larga já é uma grande aliada nesse ponto. E é dessa maneira, com a utilização da internet banda larga, que esse projeto demonstra o canal de retorno de uma aplicação para TVDI.

### 3.19 Normas ABNT

Em 27 de março de 2007, o Fórum Brasileiro de TV Digital solicitou à ABNT – Associação Brasileira de Normas Técnicas – através do seu Módulo Técnico, a constituição

da CEET – Comissão de Estudos Especial Temporários – de TV Digital para elaborar normas que atendam ao Sistema Brasileiro de Televisão Digital Terrestre, conforme estabelece o Decreto 5.820 de 29 de junho de 2006.

Assim, foram criados grupos de trabalho para cada área do conjunto de especificações estabelecidas pelo Módulo Técnico, e dessa forma, as normas foram criadas:

- ✓ Transmissão:
  - ABNT NBR 15601
- ✓ Codificação:
  - ABNT NBR 15602 – Parte 1
  - ABNT NBR 15602 – Parte 2
  - ABNT NBR 15602 – Parte 3
- ✓ Multiplexação:
  - ABNT NBR 15603 – Parte 1
  - ABNT NBR 15603 – Parte 2
  - ABNT NBR 15603 – Parte 3
  
- ✓ Receptores:
  - ABNT NBR 15604
- ✓ Segurança:
  - ABNT NBR 15605
- ✓ Middleware:
  - ABNT NBR 15606 – Parte 1
  - ABNT NBR 15606 – Parte 2
  - ABNT NBR 15606 – Parte 3
  - ABNT NBR 15606 – Parte 4
  - ABNT NBR 15606 – Parte 5
- ✓ Canal de Interatividade:

- ABNT NBR 15607
- ✓ Guia de Operação:
  - ABNT NBR 15608 – Parte 1
  - ABNT NBR 15608 – Parte 2
  - ABNT NBR 15608 – Parte 3

No próximo capítulo, será exposto o modelo de proposta de projeto e vários aspectos técnicos, os quais já foram citados anteriormente, que serão úteis na implementação.

## CAPÍTULO 4 - PROPOSTA DE SOLUÇÃO

O projeto consiste em desenvolver uma solução que permita a interatividade, tendo como base a tecnologia de TV Digital, especificamente a do modelo brasileiro, para o qual serão utilizadas ferramentas como Ginga, NCL, Lua e PHP. Para atender este objetivo será criado um documento hipermídia, ou seja, um conjunto de códigos em várias linguagens com mídias diferentes sendo utilizadas, para a realização de uma enquete interativa cuja visualização será iniciada através de um ícone interativo (i) que estará disponibilizado durante toda a apresentação. Será utilizado como exemplo, um comercial do UniCEUB. Esse ícone interativo será acionado através de comandos enviados pelo set-top-box virtual (Ginga Virtual STB – Máquina Virtual) que será manipulado pelo usuário. Essa interação será uma enquete que tratará sobre a inclusão de novos cursos na instituição de ensino. A enquete trará três opções de curso – Agronomia, Moda e Música - o usuário selecionará uma dessas opções. Sua resposta será disponibilizada na tela e enviada para ele via email (canal de retorno). Dessa forma, o objetivo desse documento é o recebimento de um email com a resposta dada à enquete pelo usuário durante o comercial da instituição. Na figura 57, temos o diagrama do projeto:

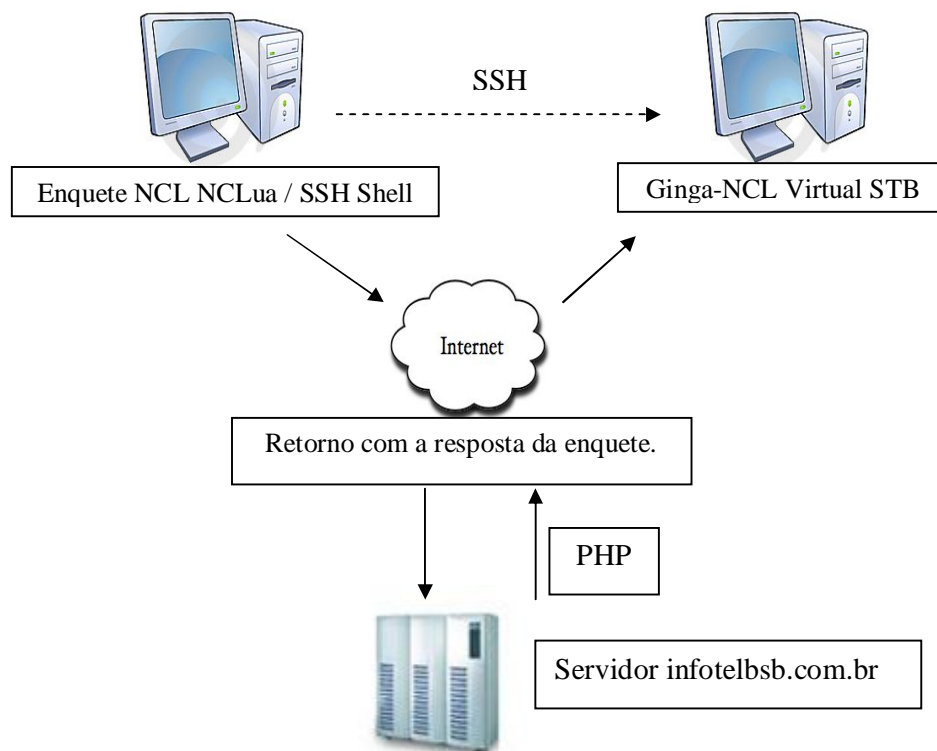


Figura 57 – Diagrama do Projeto

#### 4.1. Arquitetura do código

Abaixo, na figura 58, temos a visualização da enquete interativa com as linguagens utilizadas e a ordem em que as aplicações se conectam.

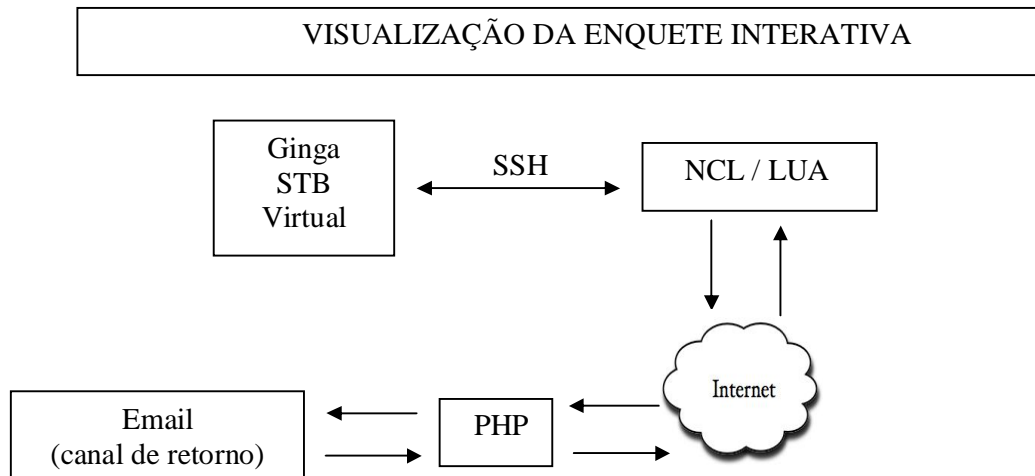


Figura 58 – Visualização da Enquete Interativa

##### 4.1.1 Produção de código

Para a criação da enquete com o canal de retorno foram necessárias três linguagens de programação: NCL, Lua e PHP.

A linguagem NCL é a linguagem principal do projeto. Este arquivo contém toda a programação necessária para disparar o evento interativo conforme desejado.

Arquivos complementares de vídeo, áudio, texto e imagens são gerenciados pelo documento NCL. Esses arquivos podem estar juntos fisicamente em um mesmo diretório ou com outro arquivo NCL os chamando por meio de um caminho físico ou virtual até onde se encontram.

No caso da enquete interativa do comercial do UniCEUB, todos os arquivos complementares necessários para a execução local estão juntos com o código NCL, organizados em um diretório principal e um subdiretório para as mídias utilizadas dentro dele. O diretório principal contém em sua raiz os arquivos *causalConnBase.ncl* e *projfinal.ncl*, além do diretório de mídias *mediaceub*. O arquivo *projfinal.ncl* é o arquivo principal.

A organização desta forma deixa a aplicação independente para ser executada em qualquer ambiente que respeite o SBTVD.

Nesse desenvolvimento, a linguagem Lua também foi utilizada. Um código NCLua deve estar em outro arquivo referenciado pelo arquivo NCL. O ciclo de vida do arquivo Lua é definido pelo arquivo NCL que o chama.

PHP, *Hypertext Preprocesso* é uma linguagem de programação muito utilizada para gerar conteúdo dinâmico na internet. É uma linguagem de domínio específico, ou seja, seu escopo se estende a um campo de atuação que é o desenvolvimento web. Com isso, nesse projeto, ela fica responsável pelo envio do email após a digitação do respectivo endereço, feito por Lua. Na figura 59, temos a representação da escrita do código, com as linguagens:

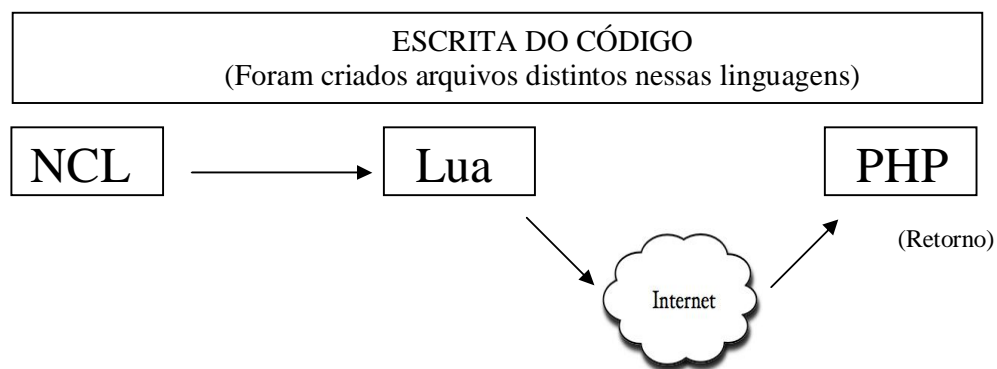


Figura 59 – Arquitetura do código do projeto

Para a compilação desses arquivos, foram utilizadas algumas IDEs (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento), que são programas que reúnem ferramentas para o desenvolvimento de softwares específicos em uma determinada linguagem.

Para os arquivos NCL (.ncl), foi utilizado o Eclipse SDK, versão 3.5.1. Assim foram criados os dois arquivos ncl em uso: *projfinal*. e o *causalConnBase.ncl*.

Para os arquivos Lua (.lua), foi utilizado o Scite, versão 1.75. Assim, foram criados os seguintes arquivos: *conecta.lua*, *tcp.lua*, *email\_agronomia.lua*, *email\_moda.lua*, *email\_musica.lua*.

Para os arquivos PHP (.php), foi utilizado o Notepad, renomeando-se as suas extensões e enviados para o servidor via ftp, utilizando o software FileZilla, versão 3.2.8.1.



## 4.2 O Código

Esse item traz um maior detalhamento do arquivo principal *projfinal.ncl*. É a partir desse arquivo que a enquete interativa é inicialmente demonstrada ao usuário, através do ícone interativo (i) e posteriormente acessada com a pergunta e as possíveis respostas que o usuário poderá selecionar, caso houver interesse.

O código completo, de toda a interatividade, incluindo os códigos das outras linguagens, está disponível no Apêndice A.

### 4.2.1 Documento NCL

Como foi mencionado no capítulo anterior, um arquivo NCL possui algumas estruturas pré-definidas que devem ser seguidas, como por exemplo, o cabeçalho do documento que é dividido em:

→ Estrutura inicial:

```
<?xml version="1.0" encoding="ISO-8859-1">
```

Aqui são definidas a versão do xml bem como a codificação. Isto é, define o tipo de caracteres que serão exibidos na tela. A codificação usada nesse projeto corresponde aos caracteres ocidentais, ou seja, a "ISO-8859-1". No caso da implementação feita nesse trabalho, essa estrutura pode ser vista no Apêndice A, com todo o código escrito.

→ Cabeçalho do programa:

```
<ncl id="projfinal" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
```

Essa tag define o início do programa ncl, que é conhecido como *nó ncl*, simbolizando toda a aplicação NCL. Ela possui dois atributos: 'id' e 'xmlns'. O 'id' é utilizado para identificar a aplicação, e o 'xmlns' serve para identificar o perfil NCL que está sendo usado. No caso da implementação feita nesse trabalho, essa estrutura também pode ser vista no Apêndice A, com todo o código escrito.

#### 4.2.2 Regiões

As regiões são tags que ficam dentro do cabeçalho (“head”). Elas servem para especificar o posicionamento das mídias (vídeo, texto, imagens) na tela de exibição.

No caso da implementação feita neste trabalho, foram determinadas as seguintes regiões:

- ✓ Região Mãe – rgMae: representa a região total da tela;
- ✓ Região Vídeo – rgVideo: determina a posição do vídeo;
- ✓ Região Interatividade – rgInteratividade: determina a área da enquete interativa, incluindo o logo da instituição;
- ✓ Região Tela Ajuda – rgTela\_Ajuda: determina a área ocupada da tela de ajuda;
- ✓ Região Botão Interativo – rgBotaoInterativo: indica a posição do botão (i), que demonstra o início da interatividade;
- ✓ Região Botões Abaixo – rgBotoesAbaixo: utilizada para indicar onde os botões de *Sair e Ajuda* vão ficar na tela;
- ✓ Região Botão Sair2 – rgBotaoSair2: determina a posição do botão *Sair* da tela de Ajuda;
- ✓ Região Logo Moda - rgLogo\_Moda: indica o local do botão de opção de escolha do curso Moda;
- ✓ Região Logo Musica - rgLogo\_Musica: indica o local do botão de opção de escolha do curso Música;
- ✓ Região Logo Agronomia - rgLogo\_Agronomia: indica o local do botão de opção de escolha do curso Agronomia;
- ✓ Região Email – rgEmail: determina a posição de escrita do email;
- ✓ Região Lua – regioaLua: indica o local que o arquivo Lua vai ser utilizado no NCL.

No Apêndice A, pode-se encontrar todo o código completo referente ao projeto.

### 4.2.3 Descritores

Os descritores definem como as mídias serão apresentadas, isto significa que quando formos declarar uma mídia teremos que relacioná-la a um descritor, e este descritor irá apresentar algumas características que auxiliarão na exibição da mídia, como por exemplo a cor e a espessura da borda de um botão.

No código do projeto, temos os descritores para cada uma das regiões estipuladas no item anterior. Incluindo algumas funcionalidades de descrição das mídias, como por exemplo, nas mídias das opções de cursos, a ordem do caminho de seleção de cada botão de acordo com a indicação das setas do controle remoto. Para isso, são utilizadas a função *focusIndex*, que indica em qual botão estará o foco da seleção naquele determinado momento, e as funções *moveright* e *moveleft* para indicar para qual lado o foco pode ir, já que se trata de um menu horizontal. Temos também a questão de cor e espessura das bordas de seleção, com as funções *focusBorderColor* e *focusBorderWidth*.

No Apêndice A, o código completo do projeto está disponível para consultas.

### 4.2.4. Conectores

Os conectores definem como os elos são ativados e o que eles disparam. Através de conectores, é possível fazer com que a exibição de uma mídia comece simultaneamente a outra, bem como termine simultaneamente a outra e diversas outras ações, por exemplo.

O arquivo *causalConnBase.ncl*, que está no mesmo diretório do programa ncl possui vários conectores prontos, e é carregado no arquivo principal *projfinal.ncl* através do atributo "documentURI". O atributo "alias" funciona como uma espécie de identificador para a base carregada. Essa "identificação" será utilizada pelos elos para fazer referência à base carregada.

Esse arquivo está disponível para consultas no Apêndice B.

#### 4.2.5 Nós de mídia

São os arquivos de imagem, vídeo e texto. O principal nó de mídia do projeto é o vídeo do comercial da instituição que é exibido independente do conteúdo interativo. Entretanto, foram utilizados também arquivos de imagem (.png) para a criação das telas de interatividade e ajuda, e os botões, tanto das opções de curso, quanto o de *Sair* e *Ajuda*.

Na implementação feita nesse trabalho, essa estrutura também pode ser vista no Apêndice A, com todo o código escrito para eventuais consultas.

#### 4.2.6 Links

São as ligações entre nós de mídia. Quando o usuário acessa uma das respostas da enquete interativa, por exemplo, o *link* no código vinculado à ação dessa mídia é acionado. O código completo pode ser acessado no Apêndice A.

### 4.3. Servidor Web – infotelbsb.com.br

Para que o e-mail seja enviado a partir da aplicação interativa, é necessário que haja um arquivo PHP, com a mensagem desejada e emails configurados, em algum servidor web disponível. Dessa forma, disponibilizei via FTP os arquivos *respostaagronomia.php*, *respostamoda.php* e *respostamusica.php* nesse servidor, que é o servidor da empresa Infotel – Informática e Telecomunicações, minha empresa. Esses arquivos serão chamados a partir do arquivo Lua *conecta.lua*, que determina a escolha feita pelo usuário e o email digitado.

Os arquivos php estão no Apêndice D para eventuais consultas.

## **CAPÍTULO 5 – APLICAÇÃO E DEMONSTRAÇÃO DA SOLUÇÃO COM RESULTADOS**

Este capítulo demonstra a simulação do projeto, com as telas e os resultados conforme a aplicação é executada. Aqui, teremos toda a configuração das máquinas e softwares utilizados nesse projeto.

Com o intuito de simular a recepção e visualização na casa do usuário, foi montado um ambiente utilizando dois notebooks, um simulando a emissora transmitindo o conteúdo, e o outro simulando a TV do usuário, com a visualização e acesso à enquete interativa. Para um melhor entendimento, chamaremos *NoteTX*, marcação 3 na figura 59, abaixo, o notebook responsável por enviar o aplicativo; e *NoteRX*, marcação 1, o notebook receptor. Os dois estão ligados em um roteador D-Link, marcação 2, como será melhor exposto mais a frente.

As máquinas possuem as seguintes configurações:

- ✓ NoteTX: AMD Athlon 64 X2 Dual-Core Processor TK-57, 1.90GHz, 2,5GB RAM, Windows Vista, Toshiba.
- ✓ NoteRX: Intel Core 2Duo T6500, 2.10GHz, 4GB RAM, Windows Vista, HP.



Figura 59 – Infra-estrutura para simulação do projeto

### 5.1. Instalação da Máquina Virtual

O primeiro passo é executar o Ginga STB Virtual, através da máquina virtual VMware Player, versão 3.0. A máquina virtual referente ao Ginga-NCL Set-top-box virtual é a *fedora-fc7-ginga-i386.vmx*. Essa máquina virtual foi instalada no NoteRX, já que ela simula o set-top-box que estará na casa do usuário.

Abaixo, temos o passo a passo para a sua instalação:

1º) Baixe a máquina virtual pelo Portal do Software Público Brasileiro: [www.softwarepublico.com.br](http://www.softwarepublico.com.br). Basta criar um login e senha para acesso;

2º) Após o download da máquina virtual *fedora-fc7-ginga-i386.zip*, descompacta-se o arquivo em um diretório de seu disco rígido. Abra o VMWare Player, clique em "open" e selecione o arquivo *fedora-fc7-ginga-i386.vmx* no diretório onde você descompactou sua máquina virtual. Veja a figura 60:

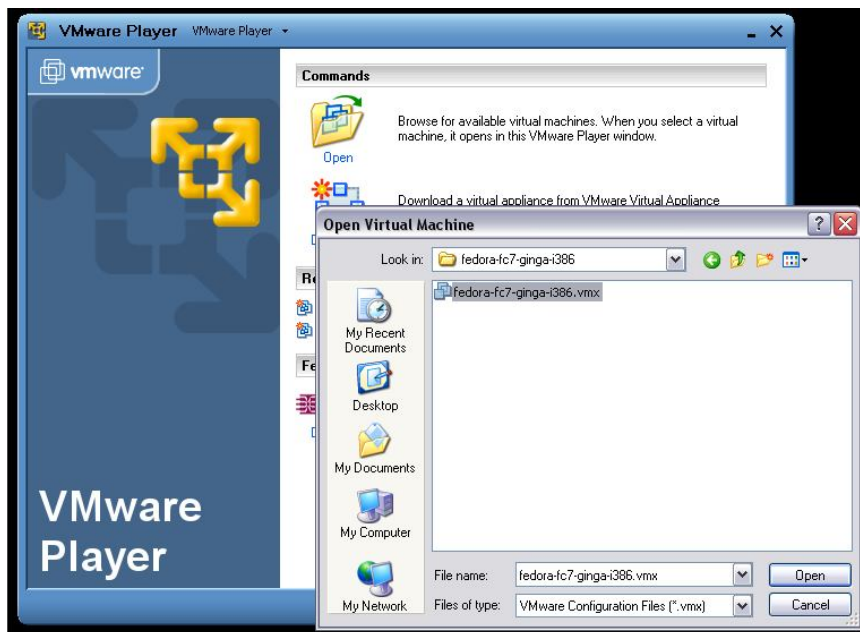


Figura 60 – Selecionando a máquina virtual do Ginga-NCL

3º) Ao abrir a máquina virtual, um processo de boot é iniciado. Não modificar a seleção default do boot loader (grub). Assim, o kernel e toda a configuração gráfica do set-top box serão carregados. A seleção default é "Ginga-NCL Development Set-top Box - v.x.x.xx", como mostra a figura 61.

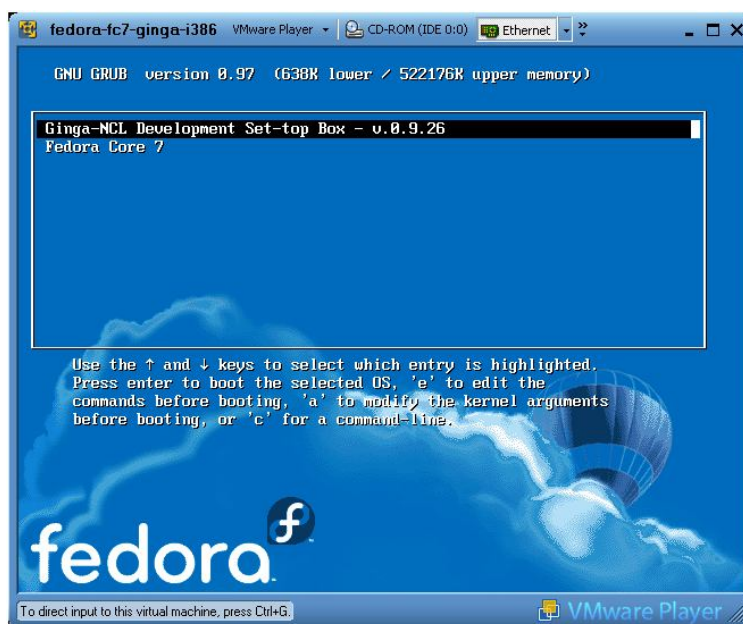


Figura 61 – Seleção default: "Ginga-NCL Development Set-top-box"

4º) Após selecionar o Ginga-NCL, a máquina virtual estará pronta para ser usada e a tela inicial é idêntica a esta abaixo:

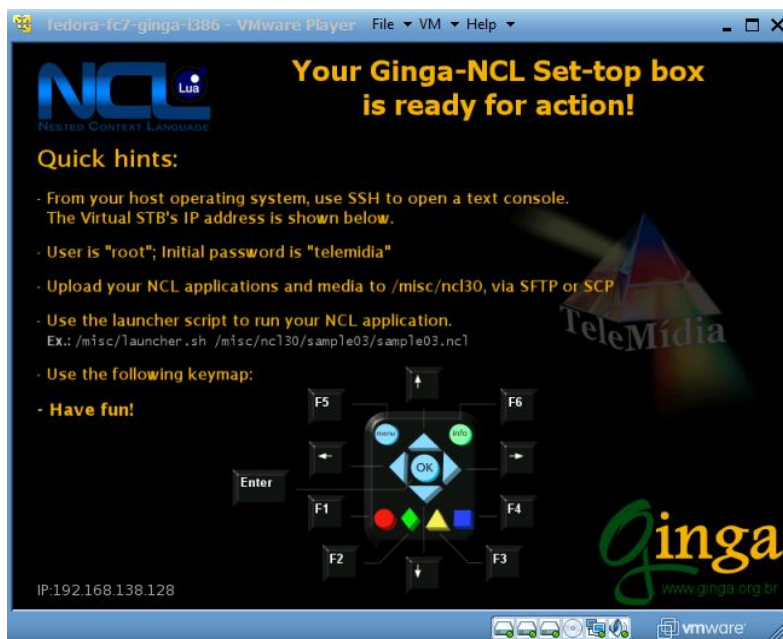


Figura 62 – Tela inicial da máquina virtual Ginga-NCL

Nessa tela, temos algumas informações como:

- ✓ O endereço IP automaticamente associado ao set-top box virtual (no canto inferior esquerdo);
- ✓ O mapeamento usado pelo Ginga-NCL para as teclas do controle remoto sobre o teclado do PC:
  - [F1] - Botão Vermelho;
  - [F2] - Botão Verde;
  - [F3] - Botão Amarelo;
  - [F4] - Botão Azul;
  - [F5] - Botão Menu;
  - [F6] - Botão Info;
  - [Enter] - Botão OK;
  - [Setas] - Botões de setas direcionais;
  - [Números] - Botões numéricos.



## 5.2. Execução e resultados da interatividade

Para a comunicação, foi utilizado o software SSH Secure Shell Client versão 3.2.9. Esse software foi instalado no NoteTX, já que ele está simulando o envio do aplicativo interativo pela emissora.

Abaixo temos as telas dos ambientes.

Essa primeira tela, figura 63, demonstra o início da conexão do SSH com o Ginga-NCL set-top-box virtual. Note que a tela do set-top-box virtual traz um número de ip, ou seja, o ip da máquina virtual que vamos acessar via SSH.

Para acessarmos, temos que preencher os seguintes dados na tela do SSH:

- ✓ **Host Name:** 192.168.0.101 (O endereço IP impresso na interface gráfica do Set-top Box Virtual);
- ✓ **User Name:** root;
- ✓ **Port Number:** 22 (é a porta default);
- ✓ **Authentication Method:** <Profile Settings> (deixe também a configuração default) ;
- ✓ Após conectar, a senha solicitada é *telemidia*.

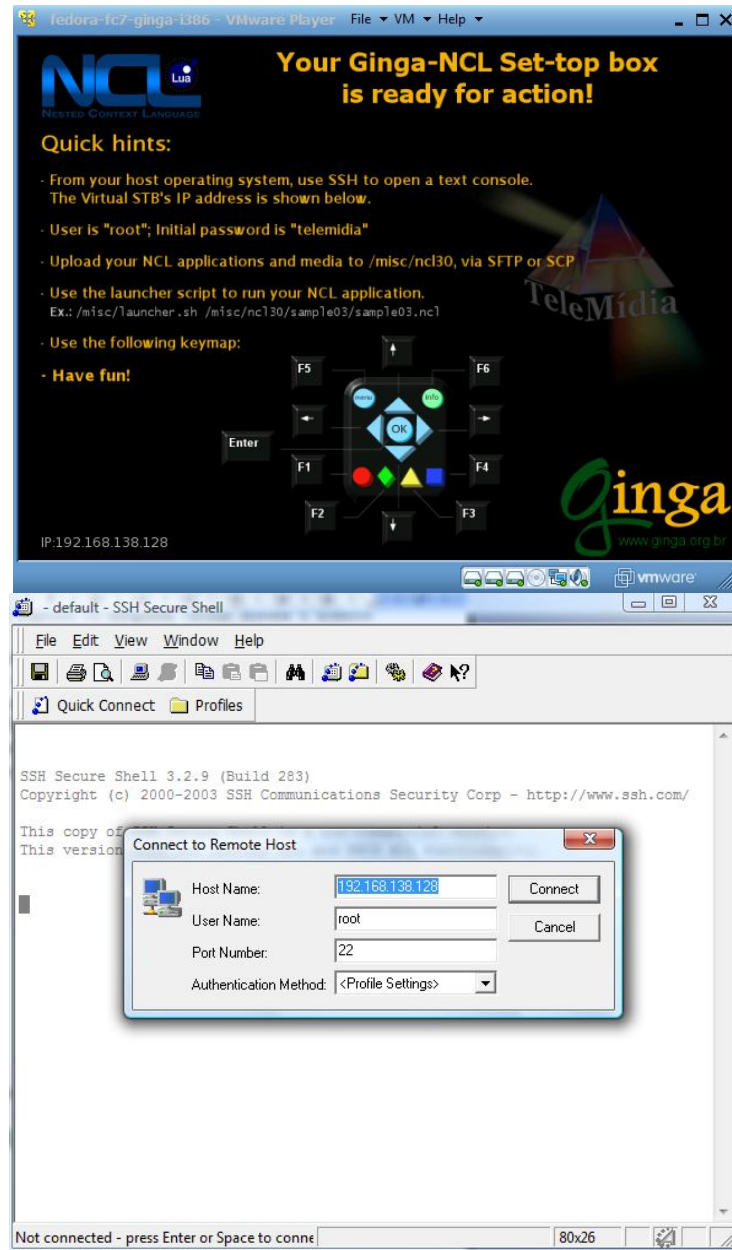


Figura 63 – Telas para execução do set-top-box virtual via SSH

Agora o console remoto está pronto para a entrada de comandos, como mostra a figura63, abaixo:

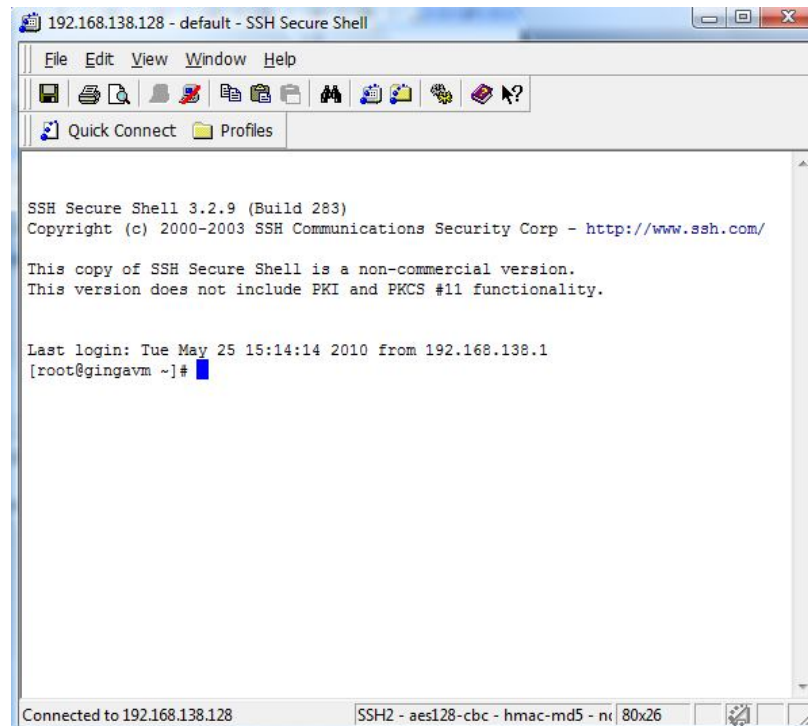


Figura 64 – Console SSH

No Set-top Box Virtual, o diretório onde armazenamos as aplicações NCL é o */misc/ncl30*. Para execução da aplicação do projeto, basta digitar o comando:

```
[root@gingavm~]# /misc/launcher.sh /misc/ncl30/projetofinal/projfinal.ncl
```

Ao iniciar a execução, o comercial interativo do UniCEUB irá disparar no set-top-box virtual, simulando uma televisão. Lembrando que nesse projeto estamos utilizando um vídeo editado para a demonstração. No caso real, esse vídeo seria a própria programação que a emissora estivesse transmitindo. Com isso, podemos comprovar que a interatividade é independente do tempo do comercial, por exemplo. Caso o comercial da instituição terminasse, o usuário ainda estaria acessando a aplicação e realizando a sua escolha.



Figura 65 – Execução do comercial interativo do UniCEUB

Note que, o ícone interativo (i) está no canto superior esquerdo do vídeo, indicando que aquele comercial possui uma interatividade disponível para o usuário (telespectador). Caso o usuário tenha interesse em acessar essa interatividade, basta clicar no botão vermelho do controle remoto, caso estivéssemos simulando com um set-top-box real. Na simulação aqui realizada, o usuário pode acessar a interatividade pressionando a tecla [F1], correspondente ao botão vermelho do controle. Para isso, é necessário jogar o foco para a máquina virtual, bastando clicar na interface gráfica do set-top-box virtual. Para voltar o foco para a interface local, o próprio computador, basta pressionar as teclas <Ctrl>+<Alt>.

Após clicarmos [F1], o vídeo do comercial é reduzido no canto superior direito para que a tela da interatividade com a enquete possa ser demonstrada:



Figura 66 – Tela da interatividade do comercial interativo do UniCEUB

A tela acima, figura 66, mostra além da pergunta e das opções da enquete: “*Qual dos cursos abaixo você gostaria que o UniCEUB oferecesse?*”, mais dois botões no canto inferior direito: “*Sair*” e “*Ajuda*”.

Clicando novamente no botão vermelho do controle remoto, caso fosse no set-top-box real ou no [F1], o usuário estaria voltando para o vídeo do comercial em tela cheia, ou estaria vendo a programação da emissora novamente.

Caso ele precise de ajuda, ele clicará no botão amarelo do controle remoto; na simulação aqui realizada, apertará a tecla [F3] para que uma tela de ajuda seja aberta, sobrepondo inclusive ao vídeo. Veja abaixo, na figura 67:



Figura 67 – Tela de ajuda da interatividade

Para voltar para a enquete, basta aperta a tecla [F3], no caso do projeto, ou o botão verde do controle remoto, como indicado na figura 67, acima.

Voltando à enquete, o usuário possui três opções de cursos novos:

- ✓ Design de moda;
- ✓ Música;
- ✓ Agronomia.

O usuário pode fazer a escolha do curso através das setas do controle remoto, caso estivéssemos simulando com um set-top-box real. No caso do projeto, basta o usuário navegar pelas opções utilizando as setas do próprio teclado do computador.

Supondo que o usuário decidiu escolher o curso de *Agronomia*, a tela será essa abaixo, com o curso de *Agronomia* selecionado:



Figura 68 – Tela – Curso de Agronomia selecionado na enquete interativa

Para ser feita a escolha, basta o usuário pressionar a tecla [OK] do controle remoto, caso fosse realizado com um set-top-box real. No projeto, basta pressionar a tecla [Enter] do próprio teclado.

Ao pressionar [Enter], o email do usuário será pedido, como mostra abaixo a figura 69:

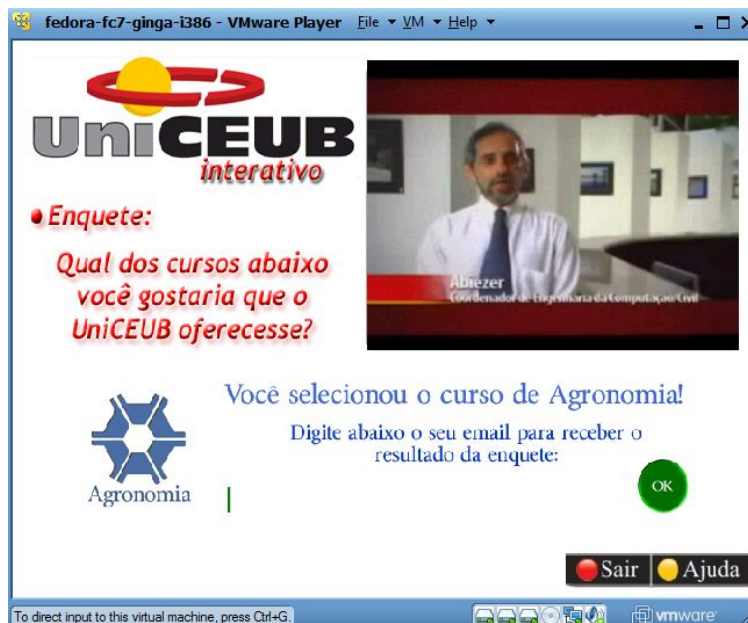


Figura 69 – Tela – Espaço para a digitação do email do usuário

Assim que o usuário digitar o email dele, como mostra a figura 70, abaixo, basta pressionar a tecla [OK] do controle remoto, caso fosse realizado com um set-top-box real, ou no caso desse projeto, a tecla [Enter] do próprio teclado, para que essa resposta seja enviada para o respectivo email, demonstrando assim, a implementação e o uso do canal de retorno com interatividade, objetivo principal desse projeto.



Figura 70 – Tela - Exemplo de email do usuário

Dessa forma, após alguns minutos, o usuário receberá a seguinte resposta no seu email:



Figura 71 – Recebimento do email pela enquete interativa do UniCEUB



Esse procedimento é idêntico para as outras respostas, ou seja, caso o usuário selecionar uma das outras opções – Moda ou Música – o processo será o mesmo, a única diferença será na resposta recebida no email, pois trará a opção escolhida.

### **5.3. A comunicação entre máquinas**

Foi utilizado um roteador D-Link DI-524 para a comunicação entre as máquinas, *NoteRX* e *NoteTX*. O roteador é um dispositivo que opera na camada 3 do modelo OSI de referência (TANEMBAUM, 2003). A principal característica desse equipamento é selecionar a rota mais apropriada para repassar os pacotes recebidos.

A conexão do roteador, marcação 2 na figura 59, simula a transmissão realizada pela emissora. A conexão entre o *NoteRX* e o GINGA-NCL virtual, via SSH representa o carrossel de objetos, enquanto o canal de retorno é demonstrado via internet.

## CAPÍTULO 6 - CONCLUSÃO

De acordo com o estudo realizado, o ISDB-T<sub>B</sub>, padrão do Sistema Brasileiro de TV Digital é, atualmente, o mais avançado sistema de TV Digital existente, não apenas por usar as tecnologias mais avançadas, mas, principalmente, por dispor de tecnologias inovadoras, como é o caso do middleware Ginga, totalmente brasileiro. Além da estupenda melhora de imagens e som, a interatividade como está sendo disponibilizada nesse atual momento é um dos principais atrativos dessa nova tecnologia.

Foi proposto e verificado uma interatividade com o uso do canal de retorno via internet como sugere o modelo brasileiro. Através da inserção de aplicações produzidas nas linguagens NCL, Lua e PHP, o usuário poderá ter contato direto com a emissora, seja respondendo enquetes, dando opiniões, sugestões ou apenas se informando sobre um determinado assunto em um programa que está sendo transmitido no momento. Como sugere a implementação aqui realizada, vários aspectos foram simulados, como transmissão e o carrossel de objetos realizados pela emissora, aqui sendo feitos por SSH, e recepção e interação do usuário com todas as funções do Set-top-box, sendo simuladas pelo Ginga-NCL Virtual Set-top-box. Dessa forma, foi possível demonstrar a interatividade.

Essa nova forma de ver e de fazer televisão é uma enorme mudança de paradigma, pois as emissoras terão que alterar toda a sua forma de apresentação dos programas e os usuários, não mais conhecidos como telespectadores, terão um canal direto com as emissoras, caso essa esteja realmente voltada para a interatividade. A televisão deixará de ser apenas um aparelho de recebimento de informações e passará a devolver respostas dos usuários para as emissoras.

A criação do comercial do UniCEUB foi apenas um exemplo na tentativa de demonstrar que qualquer instituição e/ou empresa também poderá estar diretamente ligada aos usuários de TV Digital, bastando apenas criar aplicações interativas e entrar em contato com as emissoras de sua escolha para que elas sejam transmitidas. Isso também traz uma nova forma de publicidade, alterando todo um costume já existente entre publicitários e emissoras.

Dessa forma, durante esse trabalho, muitas disciplinas estudadas durante o curso foram utilizadas e os estudos propostos são de grande valia, tanto para concretização do conhecimento adquirido ao longo do curso, bem como para atualização e expansão de

conhecimentos, uma vez que a TV Digital está em pleno processo de expansão em todo o mundo.

### **6.1 Sugestões Para Trabalhos Futuros**

Durante o decorrer de toda a implementação, inúmeras possibilidades de novas interações foram surgindo como idéias para trabalhos futuros. Segue uma lista delas:

- ✓ Verificação do resultado do vestibular, mediante entrada de número de inscrição;
- ✓ Recebimento do resultado do vestibular por SMS;
- ✓ Verificação de notas e freqüências dos alunos, mediante entrada do RA – Registro Acadêmico;
- ✓ Opinião sobre novos cursos sem opções pré-definidas, bastando o usuário enviar sua sugestão;
- ✓ Criação de mais menus interativos com todas as informações sobre o UniCEUB;
- ✓ Consulta a acervo de livros na biblioteca e posteriormente, reserva, caso o usuário seja aluno ou funcionário.
- ✓ Como o Ginga-J foi normatizado, criação de novas interatividades utilizando a linguagem JAVA.

## REFERÊNCIA BIBLIOGRÁFICA

ABNT. ABNT NBR 15601-1, Televisão digital terrestre — Sistema de Transmissão. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15602-1, Codificação de vídeo, áudio e multiplexação Parte 1: Codificação de vídeo. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15602-2, Codificação de vídeo, áudio e multiplexação Parte 2: Codificação de áudio. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15602-3, Codificação de vídeo, áudio e multiplexação Parte 3: Sistemas de multiplexação de sinais. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15603-1, Multiplexação e serviços de informação (SI) Parte 1: SI do sistema de radiodifusão. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15603-2, Multiplexação e serviços de informação (SI) Parte 2: Estrutura de dados e definições da informação básica de SI. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15603-3, Multiplexação e serviços de informação (SI) Parte 3: Sintaxes e definições de informação estendida do SI. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15604, Receptores. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15605-1, Tópicos de Segurança Parte 1: Controle de cópias. ABNT, primeira edição, 2008.

\_\_\_\_\_. ABNT NBR 15606-1, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 1: Codificação de dados. ABNT, segunda edição, 2010.

\_\_\_\_\_. ABNT NBR 15606-2, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15606-3, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 3: Especificação de transmissão de dados. ABNT, primeira edição, 2007.

\_\_\_\_\_. ABNT NBR 15606-4, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 4: Ginga-J – Ambiente para execução de aplicações procedurais. ABNT, primeira edição, 2010.

\_\_\_\_\_. ABNT NBR 15606-5, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 5: Ginga-NCL para receptores portáteis – Linguagem de aplicação XML para codificação de aplicações. ABNT, primeira edição, 2008.

\_\_\_\_\_. ABNT NBR 15607-1, Canal de Interatividade, Parte 1: Protocolos, interfaces físicas e interfaces de software. ABNT, primeira edição, 2008.

\_\_\_\_\_. ABNT NBR 15608-1, Guia de Operação, Parte 1: Sistema de transmissão – Guia para implementação da ABNT NBR 15601:2007. ABNT, primeira edição, 2008.

\_\_\_\_\_. ABNT NBR 15608-2, Guia de Operação, Parte 2: Codificação de vídeo, áudio e multiplexação – Guia para implementação da ABNT NBR 15602:2007. ABNT, primeira edição, 2008.

\_\_\_\_\_. ABNT NBR 15608-3, Guia de Operação, Parte 3: Multiplexação e serviço de informação (SI) – Guia para implementação da ABNT NBR 15603:2007. ABNT, primeira edição, 2008.

ARIB. ARIB Standard. STD-B31. Transmission System for Digital Terrestrial Television Broadcasting. Ver. 1.5. Jul. 2003.

\_\_\_\_\_. *Association of Radio Industries and Businesses*, 1997. Especificação e documentação do padrão ARIB. Disponível em: <http://www.arib.or.jp/english>. Acessado em 18 mar. 2010.

\_\_\_\_\_. STD-B24 Version 1.6. *Reference Model for Data Broadcasting. English Translation*. ARIB Standard, 30 nov. 2007.

\_\_\_\_\_. STD-B31 Version 1.6. *Transmission System For Digital Terrestrial Television Broadcasting. English Translation*. ARIB Standard, 30 nov. 2005.

ATSC. *Advanced Television Systems Committee Inc*, 2006. Especificação e documentação do padrão ATSC. Disponível em: <http://www.atsc.org/standards.html>. Acessado em 18 mar. 2010.

\_\_\_\_\_. Standard A/53. ATSC Digital Television Standard, 1995

BASTOS, Arilson e FERNANDES, Sérgio. *Televisão Digital*. 1. ed. Rio de Janeiro. Antenna Edições Técnicas Ltda, 2004.

BASTOS, Arilson e FERNANDES, Sérgio. *Televisão Profissional*. 1. ed. Rio de Janeiro. Antenna Edições Técnicas Ltda, 2003.

BBC. Digital TV. Disponível em < <http://www.bbc.co.uk/digital/tv/>>. Acessado em: 20 mar. 2010.

BEZERRA, Romildo Martins: Transmissão Digital e Analógica, Redes de Computadores I, CEFET/BA, 2008.

BITTENCOUT, Fábio Antônio; BENNERT, Wagner Alves. TV Digital: Uma análise das Modulações e das Codificações de Áudio e Vídeo no Modelo Terrestre. Universidade Tecnológica Federal do Paraná, departamento acadêmico de eletrônica tecnologia em eletrônica, Curitiba, 2007.

BRASIL. Decreto n.º 5.820, de 29 de Junho de 2006. Dispõe sobre a implantação do SBTVD-T, estabelece diretrizes para a transição do sistema de transmissão analógica para o sistema de transmissão digital do serviço de radiodifusão de sons e imagens e do serviço de retransmissão de televisão, e dá outras providências. Diário Oficial da União, República Federativa do Brasil, Poder Executivo, Brasília, DF, 30 jun. 2006.

BRASILEIRO, Portal do Software Público. How-to – Set-top Box Virtual Ginga-NCL. Disponível em

<[http://www.softwarepublico.gov.br/dotlrn/clubs/ginga/gingancl/xowiki/gingancl\\_vm](http://www.softwarepublico.gov.br/dotlrn/clubs/ginga/gingancl/xowiki/gingancl_vm)>.

Acessado em 10 abr. 2010.

CPqD. Arquitetura de Referência. OS 40.541. Campinas-SP, 2005.

DASE. DTV application software environment level 1. Part 1: introduction, architecture, and common facilities. ATSC Standard Advanced Television, 9 mar. 2003.

DEE-LUCAS, D. 1996. Effects of Overview Structure on Study Strategies and text Representations for Instructional Hypertext. In: Rouet et alii (eds), pp. 73- 107. Diferença entre interação e interatividade.

DIGITAL, Nova TV. História da TV. Disponível em:

< [http://www.novatvdigital.com/canal\\_tvhistoria.html](http://www.novatvdigital.com/canal_tvhistoria.html)>. Acessado em: 03 mar. 2010

DVB. *Digital Video Broadcasting Project*. 2003. Site oficial do projeto DVB. Disponível em: <<http://www.dvb.org>>. Acessado em: 18 mar. 2010.

DTV, Status. Mapa de Comparação dos sistemas de TV Digital no mundo. Disponível em: <<http://en.dtvstatus.net/#dtv>>. Acessado em 19 mai. 2010.

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

GINGA. Sobre o Ginga. Ginga, 2006. Disponível em < <http://www.ginga.org.br/sobre.html>>. Acessado em 26 mar. 2010.

IBGE. Pesquisa Nacional por Amostra de Domicílios 2005. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2008.

ISO. 13818-6 *Information Technology – Generic coding of moving pictures and associated audio information – Part 6 : Extensions for DSM-CC*. ISO/IEC International Organization for Standardization / Internacional Eletrotecnical Committee, 1998.

IT, Image Group. Introdução às normas propostas pelo DVB. Disponível em: < [http://www.img.lx.it.pt/~fp/cav/ano2006\\_2007/MEEC/Trab\\_22/website\\_cav/svbs.htm](http://www.img.lx.it.pt/~fp/cav/ano2006_2007/MEEC/Trab_22/website_cav/svbs.htm)>. Acessado em: 02 jun. 2010.

ITU-R. ITU-R BS.775-1,. *Multichannel Stereophonic Sound System with and without accompanying picture*. International Telecommunications Union, 1994.

MENDES, Luciano Leonel. SBTVD: Uma visão sobre a TV digital no Brasil. T&C Amazônica, ano V, número 12, out. 2007.

MHP. Digital Vídeo Broadcasting Multimedia Home Plataform. 2003. Site oficial da plataforma MHP. Disponível em: <<http://www.mhp.org>>. Acessado em: 18 mar. 2010.

MINICOM. Portaria N° 652, de 10 de outubro de 2006, Brasília, 10 out. de 2006. Disponível em: <<http://www.mc.gov.br/tv-digital/tv-digital/Portaria%20no%20652-%20de%2010-10-2006>>. Acesso em: 20 mar. 2010.

NETO, Vicente Soares. Telecomunicações: Sistemas de Modulação, 2003. 130p.

NETO, C.S.S; Soares, L.F.G; Rodrigues, R.F; Barbosa, S.D.J: Tutorial Construindo Programas Audiovisuais Interativos Utilizando a NCL 3.0 e a Ferramenta Composer; PUC-RJ, Telemídia, SERG; 2ª Ed, 2007.

NHK. NHK/Digital. Disponível em:

< [http://www.nhk.or.jp/digital/en/digitalbroad/04\\_5\\_inter.html](http://www.nhk.or.jp/digital/en/digitalbroad/04_5_inter.html)>. Acessado em: 19 mar. 2010.

PICCIONI, Carlos Alexandre: Dissertação para Pós-Graduação, UFSC: Modelo e Implementação de um Serviço de Datacasting para Televisão Digital, 2005.

SBTV/D, Fórum. História do Sistema Brasileiro de TV Digital Terrestre. Sistema Brasileiro de Televisão Digital, 2007. Disponível em:

< <http://www.dtv.org.br/materias.asp?menuid=3&id=5>>. Acessado em: 15 abr. 2010.

SOARES, L.F.S; Barbosa, S.D.J. Programando em NCL. 1 ed. Rio de Janeiro. Campus, 2009.

SOARES, L.F.S; Rodrigues, R.F. *Nested Context Model 3.0 Part 1 – NCM Core*, Monografias em Ciências da Computação de Informática, PUC-Rio, No. 18/05. Rio de Janeiro, mai. 2005.

TELECO. Cronograma de Implantação TV Digital no Mundo. Teleco, 24 abr. 2009. Disponível em: <[http://www.teleco.com.br/tvdigital\\_mundo.asp](http://www.teleco.com.br/tvdigital_mundo.asp)>. Acessado em: 28 abr. 2010.

TELEMÍDIA. Ambiente para Desenvolvimento de Aplicações Declarativas para a TV Digital Brasileira. Laboratório Telemídia, Depto. Informática, PUC-Rio Rio de Janeiro, 2006.

WIKIPÉDIA. NTSC. Mapa com o sistema de codificação televisiva por nações. Disponível em: < <http://pt.wikipedia.org/wiki/NTSC>>. Acessado em: 25 abr. 2010.

WIKIPÉDIA. 1-Seg. Disponível em: < <http://pt.wikipedia.org/wiki/1SEG>>. Acessado em: 14 abr. 2010.



## APÊNDICE A – CÓDIGO FONTE ENQUETE INTERATIVO

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="projfinal" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">

<head>

    <regionBase>
        <region width="100.0%" height="100.0%" id="rgMae">
            <region width="100.0%" height="100.0%" id="rgVideo" zIndex="1"/>
            <region width="100.0%" height="100.0%" id="rgInteratividade"/> <!--
TELA INTERATIVIDADE -->
            <region width="100.0%" height="100.0%" id="rgTela_Ajuda"
zIndex="1"/> <!-- TELA AJUDA -->
            <region left="4" top="15" width="6.0%" height="7.3%"
id="rgBotaoInterativo" zIndex="2"/> <!-- BOTÃO INTERATIVO "i" -->
            <region left="476" top="442" width="158" height="30"
id="rgBotoesAbaixo" zIndex="2"/> <!-- BOTÕES ABAIXO -->
            <region left="560" top="448" width="56" height="24"
id="rgBotaoSair2" zIndex="2"/> <!-- BOTÃO SAIR2 -->
            <region left="50" top="279" width="125" height="140"
id="rgLogo_Moda" zIndex="2"/> <!-- CURSO MODA -->
            <region left="206" top="279" width="124" height="140"
id="rgLogo_Musica" zIndex="2"/> <!-- CURSO MÚSICA -->
            <region left="363" top="279" width="124" height="140"
id="rgLogo_Agronomia" zIndex="2"/> <!-- CURSO AGRONOMIA -->
            <region left="48" top="278" width="570" height="143"
id="rgResultado_Moda" zIndex="2"/> <!-- RESULTADO MODA -->
            <region left="48" top="278" width="570" height="143"
id="rgResultado_Musica" zIndex="2"/> <!-- RESULTADO MÚSICA -->
            <region left="48" top="278" width="570" height="143"
id="rgResultado_Agronomia" zIndex="2"/> <!-- RESULTADO AGRONOMIA -->
            <region left="38" top="278" id="rgEmail" zIndex="2"/> <!-- ESPAÇO
PARA EMAIL -->
            <region id="regiaoLua" width="362" height="34" top="380"
left="180" zIndex="6" />
        </region>
    </regionBase>

```

```

<descriptorBase>
  <descriptor region="rgMae" id="dMae"/>
  <descriptor region="rgVideo" id="dVideo">
    <descriptorParam name="SoundLevel" value="90.0%"/>
  </descriptor>
  <descriptor region="rgInteratividade" id="dInteratividade"/> <!-- TELA
INTERATIVIDADE -->
  <descriptor region="rgTela_Ajuda" id="dTela_Ajuda"/> <!-- TELA AJUDA -
->
  <descriptor region="rgBotaoInterativo" id="dBotaoInterativo"/> <!-- BOTÃO
INTERATIVO "i" -->
  <descriptor region="rgBotoesAbaixo" id="dBotoesAbaixo"/> <!-- BOTÃO
ABAIXO -->
  <descriptor region="rgBotaoSair2" id="dBotaoSair2"/> <!-- BOTÃO SAIR2 --
>
  <descriptor region="rgLogo_Moda" id="dLogo_Moda" focusIndex="1"
moveRight="2" focusBorderWidth="-5" focusBorderColor="gray"/> <!-- CURSO MODA --
>
  <descriptor region="rgLogo_Musica" id="dLogo_Musica" focusIndex="2"
moveRight="3" moveLeft="1" focusBorderWidth="-5" focusBorderColor="gray"/> <!--
CURSO MÚSICA -->
  <descriptor region="rgLogo_Agronomia" id="dLogo_Agronomia"
focusIndex="3" moveLeft="2" focusBorderWidth="-5" focusBorderColor="gray"/> <!--
CURSO AGRONOMIA -->
  <descriptor region="rgResultado_Moda" id="dResultado_Moda"/> <!--
RESULTADO MODA -->
  <descriptor region="rgResultado_Musica" id="dResultado_Musica"/> <!--
RESULTADO MÚSICA -->
  <descriptor region="rgResultado_Agronomia" id="dResultado_Agronomia"/>
<!-- RESULTADO AGRONOMIA -->
  <descriptor region="rgEmail" id="dEmail"/> <!-- Arquivo TextField em Lua -
->
  <descriptor id="descritorLua" region="regiaoLua" focusIndex="ixLua" />
</descriptorBase>

<connectorBase>
  <importBase documentURI="causalConnBase.ncl" alias="connBase"/>

  <causalConnector id="onSelectionSetNStartNStopN">

```

```

    <connectorParam name="var"/>
    <simpleCondition role="onSelection"/>
    <compoundAction operator="seq">
      <simpleAction    role="set"    value="$var"    max="unbounded"
qualifier="par"/>
        <simpleAction role="start" max="unbounded" qualifier="par"/>
        <simpleAction role="stop"  max="unbounded" qualifier="par"/>
    </compoundAction>
  </causalConnector>

  <causalConnector id="onSelectionSetNStartNStopNSetN">
    <connectorParam name="var"/>
    <simpleCondition role="onSelection"/>
    <compoundAction operator="seq">
      <simpleAction    role="set"    value="$var"    max="unbounded"
qualifier="par"/>
        <simpleAction role="start" max="unbounded" qualifier="par"/>
        <simpleAction role="stop"  max="unbounded" qualifier="par"/>
    </compoundAction>
  </causalConnector>

  <causalConnector id="onKeySelectionSetNStartNStopNAbortN">
    <connectorParam name="keyCode"/>
    <connectorParam name="var"/>
    <simpleCondition role="onSelection" key="$keyCode"/>
    <compoundAction operator="seq">
      <simpleAction    role="set"    value="$var"    max="unbounded"
qualifier="par"/>
        <simpleAction    role="start"    max="unbounded"
qualifier="par"/>
        <simpleAction    role="stop"     max="unbounded"
qualifier="par"/>
        <simpleAction    role="abort"    max="unbounded"
qualifier="par"/>
    </compoundAction>
  </causalConnector>

```

```

    <causalConnector id="naSelecaoSetarPararIniciar">
    <connectorParam name="var"/>
    <simpleCondition role="onSelection"/>
    <compoundAction operator="seq">
        <simpleAction role="set" value="$var" max="unbounded"
qualifier="par"/>
        <simpleAction role="stop" max="unbounded" qualifier="par"/>
        <simpleAction role="start" max="unbounded" qualifier="par"/>
    </compoundAction>
</causalConnector>

```

```

<causalConnector id="onEndAttributionStartNStopN">
    <connectorParam name="var"/>
    <simpleCondition role="onEndAttribution"/>
    <compoundAction operator="seq">
        <simpleAction role="start" max="unbounded"
qualifier="par"/>
        <simpleAction role="stop" max="unbounded"
qualifier="par"/>
    </compoundAction>
</causalConnector>

```

```

<causalConnector id="onEndStartNStopN">
    <simpleCondition role="onEnd"/>
    <compoundAction operator="seq">
        <simpleAction role="start" max="unbounded"
qualifier="par"/>
        <simpleAction role="stop" max="unbounded"
qualifier="par"/>
    </compoundAction>
</causalConnector>

```

```

<causalConnector id="aoIniciarSetar">
<connectorParam name="var"/>
<simpleCondition role="onBegin" />
    <simpleAction role="set" value="$var" max="unbounded" qualifier="par"/>
</causalConnector>

```

```

    </connectorBase>

</head>

<body>
    <port component="video" id="entryport"/>

    <media      descriptor="dVideo"      src="mediaceub/comercial_uniceub.mpg"
type="video/mpeg" id="video">
        <property name="bounds"/>
        <property name="SoundLevel"/>
    </media>

    <media type="application/x-ginga-settings" id="programSettings">
        <property name="currentKeyMaster" value="ixLua"/>
    </media>

    <media      id="email_agronomia"      type="application/x-ginga-NCLua"
src="email_agronomia.lua"  descriptor="descriptorLua" >
        <property name="encerra" />
    </media>

    <media      id="email_moda"          type="application/x-ginga-NCLua"
src="email_moda.lua"  descriptor="descriptorLua" >
        <property name="encerra" />
    </media>

    <media      id="email_musica"        type="application/x-ginga-NCLua"
src="email_musica.lua"  descriptor="descriptorLua" >
        <property name="encerra" />
    </media>

    <media type="application/x-ginga-settings" id="settings">
        <property name="service.currentKeyMaster" />
        <property name="service.currentFocus" />

```

```

        <property name="channel.keyCapture" value="true" />
    </media>

    <media descriptor="dInteratividade" src="mediaceub/tela_interatividade.png"
type="image/png" id="interatividade"/>
        <media descriptor="dBotaoInterativo"
src="mediaceub/botao_interatividade2.png" type="image/png" id="botaointeratividade"/>
        <media descriptor="dBotoesAbaixo" src="mediaceub/botoesabaixo.png"
type="image/png" id="botoesabaixo"/>
        <media descriptor="dBotaoSair2" src="mediaceub/botao_sair2.png"
type="image/png" id="botaosair2"/>
        <media descriptor="dLogo_Moda" src="mediaceub/logo_moda.png"
type="image/png" id="Logo_Moda"/>
        <media descriptor="dLogo_Musica" src="mediaceub/logo_musica.png"
type="image/png" id="Logo_Musica"/>
        <media descriptor="dLogo_Agronomia" src="mediaceub/logo_agronomia.png"
type="image/png" id="Logo_Agronomia"/>
        <media descriptor="dTela_Ajuda" src="mediaceub/tela_ajuda.png"
type="image/png" id="Tela_Ajuda"/>
        <media descriptor="dResultado_Moda" src="mediaceub/resultado_moda.png"
type="image/png" id="Resultado_Moda"/>
        <media descriptor="dResultado_Musica" src="mediaceub/resultado_musica.png"
type="image/png" id="Resultado_Musica"/>
        <media descriptor="dResultado_Agronomia"
src="mediaceub/resultado_agronomia.png" type="image/png" id="Resultado_Agronomia"/>

    <link xconnector="connBase#onBeginStartDelay" id="inicia_filme">
        <bind role="onBegin" component="video"/>
        <bind component="botaointeratividade" role="start"/>
    </link>

    <link xconnector="connBase#onKeySelectionSetStartNStopN"
id="comecainterativade">
        <bind role="onSelection" component="botaointeratividade">
            <bindParam name="keyCode" value="RED"/>
        </bind>
        <bind role="set" component="video" interface="bounds">
            <bindParam name="varSet" value="305, 17, 320, 250"/>
        </bind>
    </link>

```

```

<bind role="start" component="botoesabaixo"/>
  <bind role="start" component="interatividade"/>
  <bind role="start" component="Logo_Moda"/>
  <bind role="start" component="Logo_Musica"/>
  <bind role="start" component="Logo_Agronomia"/>
  <bind role="stop" component="botaointeratividade"/>
</link>

<link
id="finalizainterativade">
  xconnector="connBase#onKeySelectionSetStartNStopN"
  <bind role="onSelection" component="botoesabaixo">
    <bindParam name="keyCode" value="RED"/>
  </bind>
  <bind role="set" component="video" interface="bounds">
    <bindParam name="varSet" value="0, 0, 720, 480"/>
  </bind>
  <bind role="start" component="botaointeratividade"/>
  <bind role="stop" component="botoesabaixo"/>
  <bind role="stop" component="interatividade"/>
  <bind role="stop" component="Logo_Moda"/>
  <bind role="stop" component="Logo_Musica"/>
  <bind role="stop" component="Logo_Agronomia"/>
  <bind role="stop" component="Resultado_Moda"/>
  <bind role="stop" component="Resultado_Musica"/>
  <bind role="stop" component="Resultado_Agronomia"/>
  <bind role="stop" component="email_moda"/>
  <bind role="stop" component="email_agronomia"/>
  <bind role="stop" component="email_musica"/>
</link>

<link
id="comecatelaajuda">
  xconnector="connBase#onKeySelectionSetStartNStopN"
  <bind role="onSelection" component="botoesabaixo">
    <bindParam name="keyCode" value="YELLOW"/>
  </bind>
  <bind role="set" component="video" interface="bounds">

```

```

        <bindParam name="varSet" value="0, 0, 0, 0"/>
    </bind>
    <bind role="start" component="Tela_Ajuda"/>
    <bind role="start" component="botaosair2"/>
    <bind role="stop" component="botaointeratividade"/>
    <bind role="stop" component="botoesabaixo"/>
    <bind role="stop" component="interatividade"/>
    <bind role="stop" component="Logo_Moda"/>
    <bind role="stop" component="Logo_Musica"/>
    <bind role="stop" component="Logo_Agronomia"/>
    <bind role="stop" component="Resultado_Moda"/>
    <bind role="stop" component="Resultado_Musica"/>
    <bind role="stop" component="Resultado_Agronomia"/>
    <bind role="stop" component="email_moda"/>
    <bind role="stop" component="email_agronomia"/>
    <bind role="stop" component="email_musica"/>
</link>

<link
id="finalizatelaajuda"
        xconnector="connBase#onKeySelectionSetStartNStopN"
    >
    <bind role="onSelection" component="botaosair2">
        <bindParam name="keyCode" value="GREEN"/>
    </bind>
    <bind>
        <bind role="set" component="video" interface="bounds">
            <bindParam name="varSet" value="305, 17, 320, 250"/>
        </bind>
    <bind role="start" component="botoesabaixo"/>
    <bind role="start" component="interatividade"/>
    <bind role="start" component="Logo_Moda"/>
    <bind role="start" component="Logo_Musica"/>
    <bind role="start" component="Logo_Agronomia"/>
    <bind role="stop" component="Tela_Ajuda"/>
    <bind role="stop" component="botaosair2"/>
    <bind role="stop" component="botaointeratividade"/>
    <bind role="stop" component="Resultado_Moda"/>

```



```

<bind role="stop" component="Resultado_Musica"/>
<bind role="stop" component="Resultado_Agronomia"/>
<bind role="stop" component="email_moda"/>
<bind role="stop" component="email_agronomia"/>
<bind role="stop" component="email_musica"/>
</link>

<link
id="resultado_MODA" xconnector="connBase#onKeySelectionStartNStopN"
  <bind role="onSelection" component="Logo_Moda"/>
  <bind role="start" component="Resultado_Moda"/>
  <bind role="start" component="botoesabaixo"/>
  <bind role="start" component="email_moda"/>
  <bind role="stop" component="botaointeratividade"/>
  <bind role="stop" component="Logo_Moda"/>
  <bind role="stop" component="Logo_Musica"/>
  <bind role="stop" component="Logo_Agronomia"/>
  <bind role="stop" component="email_musica"/>
  <bind role="stop" component="email_agronomia"/>
</link>

<link xconnector="onEndStartNStopN" id="sai_moda" >
  <bind role="onEnd" component="email_moda" />

  <bind role="start" component="botoesabaixo"/>
  <bind role="start" component="interatividade"/>
  <bind role="start" component="Logo_Moda"/>
  <bind role="start" component="Logo_Musica"/>
  <bind role="start" component="Logo_Agronomia"/>

  <bind role="stop" component="botoesabaixo"/>
  <bind role="stop" component="Resultado_Moda"/>
  <bind role="stop" component="email_moda"/>
  <bind role="stop" component="email_musica"/>
  <bind role="stop" component="email_agronomia"/>

```

```

</link>

<link xconnector="aoIniciarSetar">
  <bind component="email_moda" role="onBegin" />

  <bind component="settings" interface="service.currentFocus"
role="set">
    <bindParam name="var" value="ixLua" />
  </bind>

  <bind component="settings" interface="service.currentKeyMaster"
role="set">
    <bindParam name="var" value="ixLua" />
  </bind>
</link>

<link id="resultado_MUSICA" xconnector="connBase#onKeySelectionStartNStopN"
  <bind role="onSelection" component="Logo_Musica"/>

  <bind role="start" component="botoesabaixo"/>
  <bind role="start" component="interatividade"/>
  <bind role="start" component="Resultado_Musica"/>
  <bind role="start" component="botoesabaixo"/>
  <bind role="start" component="email_musica"/>
  <bind role="stop" component="botaointeratividade"/>
  <bind role="stop" component="Logo_Moda"/>
  <bind role="stop" component="Logo_Musica"/>
  <bind role="stop" component="Logo_Agronomia"/>
  <bind role="stop" component="email_moda"/>
  <bind role="stop" component="email_agronomia"/>
</link>

<link xconnector="aoIniciarSetar">
  <bind component="email_musica" role="onBegin" />

```

```

role="set">
    <bind component="settings" interface="service.currentFocus"
    <bindParam name="var" value="ixLua" />
    </bind>
role="set">
    <bind component="settings" interface="service.currentKeyMaster"
    <bindParam name="var" value="ixLua" />
    </bind>
</link>

<link xconnector="onEndStartNStopN" id="sai_musica" >
    <bind role="onEnd" component="email_musica" />

    <bind role="start" component="Logo_Moda"/>
    <bind role="start" component="Logo_Musica"/>
    <bind role="start" component="Logo_Agronomia"/>

    <bind role="stop" component="botoesabaixo"/>
    <bind role="stop" component="Resultado_Musica"/>
    <bind role="stop" component="email_musica"/>
    <bind role="stop" component="email_moda"/>
    <bind role="stop" component="email_agronomia"/>
</link>

<link
id="resultado_AGRONOMIA">
    xconnector="connBase#onKeySelectionStartNStopN"
    <bind role="onSelection" component="Logo_Agronomia"/>

    <bind role="start" component="botoesabaixo"/>
    <bind role="start" component="interatividade"/>
    <bind role="start" component="Resultado_Agronomia"/>
    <bind role="start" component="botoesabaixo"/>
    <bind role="start" component="email_agronomia"/>

    <bind role="stop" component="botaointeratividade"/>
    <bind role="stop" component="Logo_Moda"/>

```

```

    <bind role="stop" component="Logo_Musica"/>
    <bind role="stop" component="Logo_Agronomia"/>
    <bind role="stop" component="email_moda"/>
    <bind role="stop" component="email_musica"/>
</link>

<link xconnector="onEndStartNStopN" id="sai_agronomia" >
    <bind role="onEnd" component="email_agronomia" />

    <bind role="stop" component="Logo_Moda"/>
    <bind role="stop" component="Logo_Musica"/>
    <bind role="stop" component="Logo_Agronomia"/>

    <bind role="stop" component="botoesabaixo"/>
    <bind role="stop" component="Resultado_Agronomia"/>
    <bind role="stop" component="email_agronomia"/>
    <bind role="stop" component="email_moda"/>
    <bind role="stop" component="email_agronomia"/>
</link>

<link xconnector="aoIniciarSetar">
    <bind component="email_agronomia" role="onBegin" />

    <bind    component="settings"    interface="service.currentFocus"
role="set">

    <bindParam name="var" value="ixLua" />
</bind>

    <bind    component="settings"    interface="service.currentKeyMaster"
role="set">

    <bindParam name="var" value="ixLua" />
</bind>
</link>

<link xconnector="connBase#onEndStop">
    <bind role="onEnd" component="video"/>

```

```
<bind role="stop" component="botaointeratividade"/>
<bind role="stop" component="botoesabaixo"/>
<bind role="stop" component="interatividade"/>
<bind role="stop" component="Logo_Moda"/>
<bind role="stop" component="Logo_Musica"/>
<bind role="stop" component="Logo_Agronomia"/>
<bind role="stop" component="Tela_Ajuda"/>
<bind role="stop" component="botaosair2"/>
<bind role="stop" component="Resultado_Moda"/>
<bind role="stop" component="Resultado_Musica"/>
<bind role="stop" component="Resultado_Agronomia"/>
<bind role="stop" component="email_moda"/>
<bind role="stop" component="email_agronomia"/>
<bind role="stop" component="email_musica"/>
</link>
</body>
</ncl>
```

## APÊNDICE B – CÓDIGO FONTE ARQUIVO CAUSALCONNBASE.NCL

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Generated by NCL Eclipse -->
<ncl id="causalConnBase" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <connectorBase>
      <causalConnector id="onBeginStart">
        <simpleCondition role="onBegin"/>
        <simpleAction role="start" max="unbounded" qualifier="par"/>
      </causalConnector>

      <causalConnector id="onBeginStartDelay">
        <connectorParam name="delay"/>
        <simpleCondition role="onBegin"/>
        <simpleAction role="start" delay="$delay" max="unbounded"
qualifier="par"/>
      </causalConnector>

      <causalConnector id="onEndStop">
        <simpleCondition role="onEnd"/>
        <simpleAction role="stop" max="unbounded" qualifier="par"/>
      </causalConnector>

      <causalConnector id="onKeySelectionStartStop">
        <connectorParam name="keyCode"/>
        <simpleCondition role="onSelection" key="$vKey"/>
        <compoundAction operator="par">
          <simpleAction role="start" max="unbounded"
qualifier="par"/>

```

```

qualifier="par"/>
    <simpleAction    role="stop"    max="unbounded"
qualifier="par"/>
    <simpleAction    role="pause"   max="unbounded"
qualifier="par"/>
    <simpleAction    role="resume"  max="unbounded"
    </compoundAction>
</causalConnector>

```

```

<causalConnector id="naSelecaoSetarPararIniciar">
  <connectorParam name="var"/>
  <simpleCondition role="onSelection"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var" max="unbounded"
qualifier="par"/>
    <simpleAction role="stop" max="unbounded"
qualifier="par"/>
    <simpleAction role="start" max="unbounded"
qualifier="par"/>
  </compoundAction>
</causalConnector>

```

```

<causalConnector id="aoIniciarSetar">
<connectorParam name="var"/>
<simpleCondition role="onBegin" />
  <simpleAction role="set" value="$var" max="unbounded" qualifier="par"/>
</causalConnector>

```

```

<causalConnector id="onEndSet">

  <connectorParam name="varSet"/>

  <simpleCondition role="onEnd"/>

  <simpleAction role="set" value="$varSet"/>
</causalConnector>

```

```

<causalConnector id="onKeySelectionSetNStartNStopN">
  <connectorParam name="keyCode" />
  <connectorParam name="var" />
  <simpleCondition role="onSelection" key="$keyCode" />
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var" max="unbounded"
qualifier="par"/>
    <simpleAction role="start" max="unbounded" qualifier="par"/>
    <simpleAction role="stop" max="unbounded" qualifier="par"/>
  </compoundAction>
</causalConnector>

<causalConnector id="onSelectionSetNStartNStopN">
  <connectorParam name="var"/>
  <simpleCondition role="onSelection"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var" max="unbounded"
qualifier="par"/>
    <simpleAction role="start" max="unbounded" qualifier="par"/>
    <simpleAction role="stop" max="unbounded" qualifier="par"/>
  </compoundAction>
</causalConnector>

<causalConnector id="onBeginSetStart">
  <connectorParam name="var"/>
  <simpleCondition role="onBegin"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var"/>
    <simpleAction role="start" max="unbounded" qualifier="par"/>
  </compoundAction>

</causalConnector>

```



```

<causalConnector id="onKeySelectionSetNStartNStopNAbortN">
  <connectorParam name="keyCode"/>
  <connectorParam name="var"/>
    <simpleCondition role="onSelection" key="$keyCode"/>
      <compoundAction operator="seq">
        <simpleAction role="set" value="$var" max="unbounded"
qualifier="par"/>
          <simpleAction role="start" max="unbounded"
qualifier="par"/>
            <simpleAction role="stop" max="unbounded"
qualifier="par"/>
              <simpleAction role="abort" max="unbounded"
qualifier="par"/>
        </compoundAction>
      </causalConnector>

```

```

<causalConnector id="onKeySelectionStartN">
  <connectorParam name="keyCode" />
  <simpleCondition role="onSelection" key="$keyCode"/>
  <simpleAction role="start" max="unbounded" qualifier="par"/>
</causalConnector>

```

```

<causalConnector id="onKeySelectionStopSetStart">
  <connectorParam name="varSet"/>
  <connectorParam name="keyCode"/>
  <simpleCondition role="onSelection" key="$keyCode"/>
  <compoundAction operator="seq">
    <simpleAction role="stop" max="unbounded" qualifier="par"/>
    <simpleAction role="set" value="$varSet"/>
    <simpleAction role="start" max="unbounded" qualifier="par"/>
  </compoundAction>
</causalConnector>

```

```

<causalConnector id="onKeySelectionSetStartNStopN">

```

```

<connectorParam name="varSet"/>
<connectorParam name="keyCode" />
<simpleCondition role="onSelection" key="$keyCode" />
<compoundAction operator="seq">
    <simpleAction role="start" max="unbounded" qualifier="par"/>
    <simpleAction role="stop" max="unbounded" qualifier="par"/>
    <simpleAction role="set" value="$varSet"/>
</compoundAction>
</causalConnector>

```

```

<causalConnector id="onKeySelectionStartNStopN">
    <connectorParam name="keyCode" />
    <simpleCondition role="onSelection" key="$keyCode" />
    <compoundAction operator="seq">
        <simpleAction role="start" max="unbounded" qualifier="seq"/>
        <simpleAction role="stop" max="unbounded" qualifier="par"/>
    </compoundAction>
</causalConnector>

```

```

<causalConnector id="onKeySelectionSet">
    <connectorParam name="keyCode"/>

```

```

<connectorParam name="varSet"/>
    <simpleCondition role="onSelection" key="$keyCode"/>

    <simpleAction role="set" value="$varSet"/>
</causalConnector>

```

```

<causalConnector id="onKeySelectionStopNStartNAbortN">
    <connectorParam name="KeyCode"/>

    <simpleCondition role="onSelection" key="$keyCode"/>
    <compoundAction operator="par">
        <simpleAction    role="stop"    max="unbounded"
qualifier="par"/>

```

```

qualifier="par"/>
    <simpleAction    role="start"    max="unbounded"
qualifier="par"/>
    <simpleAction    role="abort"    max="unbounded"
    </compoundAction>
  </causalConnector>

  <causalConnector id="onBeginVarStart">
    <compoundCondition operator="and">
      <simpleCondition role="onBegin"/>
      <assessmentStatement comparator="eq">
        <attributeAssessment                                role="var"
attributeType="nodeProperty" eventType="attribution"/>
        <valueAssessment value="true"/>
      </assessmentStatement>
    </compoundCondition>
    <simpleAction role="start"/>
  </causalConnector>

  <causalConnector id="onOrKeySelectionStopStart">
    <connectorParam name="keyCode"/>
    <simpleCondition    role="onSelection"    key="$keyCode"
max="unbounded" qualifier="or"/>
    <compoundAction operator="seq">
      <simpleAction role="stop" max="unbounded" qualifier="par"/>
      <simpleAction role="start" max="unbounded" qualifier="par"/>
    </compoundAction>
  </causalConnector>

  <causalConnector id="onBeginStartSetDelay">
    <connectorParam name="var"/>
    <connectorParam name="delay"/>
    <connectorParam name="duration"/>
    <simpleCondition role="onBegin"/>
    <compoundAction operator="seq">
      <simpleAction role="start"/>
      <simpleAction    role="set"    value="$var"    delay="$delay"
duration="$duration"/>
    </compoundAction>
  </causalConnector>

```

```

</causalConnector>
<causalConnector id="onSelectionSetStop">
  <connectorParam name="var"/>
  <simpleCondition role="onSelection"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var"/>
    <simpleAction role="stop"/>
  </compoundAction>
</causalConnector>
<causalConnector id="onOrSelectionSetStopStart">
  <connectorParam name="var"/>
  <simpleCondition role="onSelection" max="unbounded" qualifier="or"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var"/>
    <simpleAction role="stop"/>
    <simpleAction role="start"/>
  </compoundAction>
</causalConnector>

<causalConnector id="onBeginSetGet">

  <connectorParam name="varSet"/>

  <simpleCondition role="onBegin"/>
  <compoundAction operator="seq">
    <simpleAction role="set" value="$var"/>
    <simpleAction role="get" value="$var"/>
  </compoundAction>
</causalConnector>
</connectorBase>

</head>

</ncl>

```

## APÊNDICE C – CÓDIGO FONTE ARQUIVOS LUA

### 1) Arquivo *conecta.lua*

```
require 'tcp'

function conectar(email, voto)

    SERVER_IP = "www.infotelbsb.com.br"

    SERVER_PORT = 80

    tcp.connect (SERVER_IP, SERVER_PORT)

    tcp.send('GET
http://www.infotelbsb.infotelbsb.com.br/'..voto..'php?email='..email..'r\n\r\n')

    return tcp.receive('*a')

end
```

### 2 ) Arquivo *email\_agronomia.lua*

```
-- Início: Email_agronomia.lua

local DX, DY = canvas:attrSize() -- Dimensões de tela.

local DRAWS = {}                -- Lista de figuras.

local INPUT = nil              -- Entrada texto.

local EMAIL = ""               -- Variável que armazena o E-Mail digitado

local ENVIAR = nil             -- Flag que determina quando o E-Mail pode ser
                                enviado. nulo = não

local IGNORE_KEYS = true      -- Ignora entrada de eventos.

-- Compõe um objeto texto em um canvas.

local function draw_text (self)

    canvas:attrColor(self.color)

    canvas:attrFont(self.font.face, self.font.height)
```

```
    canvas:drawText (self.x, self.y, self.text)
end

-- Redesenha todas as figuras.

local function redraw ()

    -- Preenche canvas com uma cor transparente.

    canvas:attrColor (0, 0, 0, 0)
    canvas:clear ()

    for i=1, #DRAWS do
        DRAWS[i]:draw()
    end

    canvas:flush ()

end

-- Eventos do teclado.

local function handler_keypress (evt)

    if evt.class == 'key' and evt.type == 'press' then

        if enviar then
```

```
if evt.key == 'ENTER' then
    assert(coroutine.resume(APP))
end

else

if evt.key == 'ENTER' then
    enviar = 1
    assert(coroutine.resume(APP))

elseif (evt.key=="") and INPUT then

    EMAIL = EMAIL .. '@'
    redraw()
    assert(coroutine.resume(APP))

-- Deleta Último caractere.

elseif (evt.key=='CURSOR_LEFT') and INPUT then

    INPUT.text = string.sub (INPUT.text, 1, -2)
    EMAIL = string.sub(EMAIL, 1, -2)
    assert(coroutine.resume(APP))

else
```

```
        EMAIL = EMAIL .. evt.key
        assert(coroutine.resume(APP))

    end

end

end

end

-- Main.
require 'conecta'
APP = coroutine.create (
    function ()
        event.register (handler_keypress)

        INPUT = { text='|', font={ face='vera', height=22},
            color='green', x=0, y=0, draw=draw_text }

        -- Loop para digitação do e-mail
        while (true) do

            INPUT.text = EMAIL.."|"

            DRAWS[#DRAWS + 1] = INPUT

            redraw ()

            -- Espera pela tecla entrada.
            coroutine.yield ()
        end
    end
)
```



```

-- Remove o texto que estão em DRAWS.

DRAWS[#DRAWS] = nil

    -- Ao pressionar "ENTER", enviar torna-se verdadeiro
    if enviar then
        MENSAGEM = { text='Obrigado por responder a nossa enquete!',
font={face='vera', height=14},
        color='green', x=0, y=0, draw=draw_text }

        MENSAGEM2 = { text='Sua resposta serã; enviada por e-mail em breve!',
font={face='vera', height=14},
        color='green', x=0, y=16, draw=draw_text }

        DRAWS[#DRAWS + 1] = MENSAGEM
        DRAWS[#DRAWS + 1] = MENSAGEM2

    redraw ()

    -- Usa a função "conectar" em "conecta.lua", enviando o e-mail e o voto por
parâmetro
    conectar(EMAIL, 'respostaagronomia')

    coroutine.yield ()

    DRAWS[#DRAWS] = nil
    DRAWS[#DRAWS] = nil
    break

```

```
        end

    end

    -- Evento que atribui a ação "stop" Ã  propriedade "encerra", declarada no ncl
    -- na mídia "Email_agronomia". Faz com que a mídia pare na execução do ncl.
    local evt = {
        class = 'ncl',
        type = 'presentation',
        name = 'encerra',
        action = 'stop',
    }

    event.post(evt)

end)

-- Registra evento handler
event.register (
    function (evt)
        if evt.class == 'ncl' and evt.type == 'presentation' and evt.action == 'start' then
            coroutine.resume (APP)
        end
    end
end
)
```

### 3) Arquivo *email\_moda.lua*

```

local DX, DY = canvas:attrSize() -- Dimensões de tela.

local DRAWS = {}                -- Lista de figuras.

local INPUT = nil              -- Entrada texto.

local EMAIL = ""               -- Variável que armazena o E-Mail digitado

local ENVIAR = nil            -- Flag que determina quando o E-Mail pode ser
enviado. nulo = não

local IGNORE_KEYS = true      -- Ignora entrada de eventos.

-- Cria um objeto texto em um canvas.

local function draw_text (self)

    canvas:attrColor(self.color)

    canvas:attrFont(self.font.face, self.font.height)

    canvas:drawText (self.x, self.y, self.text)

end

-- Redesenha todas as figuras.

local function redraw ()

    -- Preenche canvas com uma cor transparente.

```

```
canvas:attrColor (0, 0, 0, 0)

canvas:clear ()

for i=1, #DRAWS do
    DRAWS[i]:draw()
end

canvas:flush ()

end

-- Eventos do teclado.

local function handler_keypress (evt)

    -- Retoma.

    if evt.class == 'key' and evt.type == 'press' then

        if enviar then

            if evt.key == 'ENTER' then

                assert(coroutine.resume(APP))
            end
        end
    end
end
```

```
end
```

```
else
```

```
if evt.key == 'ENTER' then
```

```
    enviar = 1
```

```
    assert(coroutine.resume(APP))
```

```
elseif (evt.key=="") and INPUT then
```

```
    EMAIL = EMAIL .. '@'
```

```
    redraw()
```

```
    assert(coroutine.resume(APP))
```

```
-- Deleta último caractere.
```

```
elseif (evt.key=='CURSOR_LEFT') and INPUT then
```

```
    INPUT.text = string.sub (INPUT.text, 1, -2)
```

```
    EMAIL = string.sub(EMAIL, 1, -2)
```

```
    assert(coroutine.resume(APP))
```

```
        else

            EMAIL = EMAIL .. evt.key

            assert(coroutine.resume(APP))

        end

    end

end

end

end

-- Main.

require 'conecta'

APP = coroutine.create (

    function ()

        event.register (handler_keypress)

        INPUT = { text='|', font={ face='vera', height=22},

            color='green', x=0, y=0, draw=draw_text }

    end

)

-- Loop para digitação do e-mail
```

```

while (true) do

    INPUT.text = EMAIL..'!'

    DRAWS[#DRAWS + 1] = INPUT

    redraw ()

    -- Espera pela tecla entrada.

    coroutine.yield ()

    -- Remove o texto que estão em DRAWS.

    DRAWS[#DRAWS] = nil

    -- Ao pressionar "ENTER", enviar torna-se verdadeiro

    if enviar then

        MENSAGEM = { text='Obrigado por responder a nossa enquete!',
font={face='vera', height=14},

        color='green', x=0, y=0, draw=draw_text }

        MENSAGEM2 = { text='Sua resposta serÃ¡ enviada por e-mail em breve!',
font={face='vera', height=14},

        color='green', x=0, y=16, draw=draw_text }

        DRAWS[#DRAWS + 1] = MENSAGEM

```

```
DRAWS[#DRAWS + 1] = MENSAGEM2

redraw ()

-- Usa a função "conectar" em "conecta.lua", enviando o e-mail e o voto por
parâmetro

conectar(EMAIL, 'respostamoda')

coroutine.yield ()

DRAWS[#DRAWS] = nil

DRAWS[#DRAWS] = nil

break

end

end

-- Evento que atribui a ação "stop" Ã propriedade "encerra", declarada no ncl
-- na mídia "Email_moda". Faz com que a mídia pare na execução do ncl.

local evt = {

class = 'ncl',
```



```
        type = 'presentation',  
        name = 'encerra',  
        action = 'stop',  
    }  
    event.post(evt)  
  
end)  
  
-- Registra evento handler  
  
event.register (  
    function (evt)  
        if evt.class == 'ncl' and evt.type == 'presentation' and evt.action == 'start' then  
            coroutine.resume (APP)  
        end  
    end  
end  
)
```

#### 4) Arquivo *email\_musica.lua*

-- Início: Email\_musica.lua

local DX, DY = canvas:attrSize() -- Dimensões de tela.

local DRAWS = {} -- Lista de figuras.

local INPUT = nil -- Entrada texto.

local EMAIL = "" -- Variável que armazena o E-Mail digitado

local ENVIAR = nil -- Flag que determina quando o E-Mail pode ser  
enviado. nulo = não

local IGNORE\_KEYS = true -- Ignora entrada de eventos.

-- Cria um objeto texto em um canvas.

local function draw\_text (self)

    canvas:attrColor(self.color)

    canvas:attrFont(self.font.face, self.font.height)

    canvas:drawText (self.x, self.y, self.text)

end

-- Redesenha todas as figuras.

local function redraw ()

```
-- Preenche canvas com uma cor transparente.
```

```
canvas:attrColor (0, 0, 0, 0)
```

```
canvas:clear ()
```

```
for i=1, #DRAWS do
```

```
    DRAWS[i]:draw()
```

```
end
```

```
canvas:flush ()
```

```
end
```

```
-- Eventos do teclado.
```

```
local function handler_keypress (evt)
```

```
    -- Retoma.
```

```
    if evt.class == 'key' and evt.type == 'press' then
```

```
        if enviar then
```

```
if evt.key == 'ENTER' then  
  
    assert(coroutine.resume(APP))  
  
end
```

```
else
```

```
if evt.key == 'ENTER' then  
  
    enviar = 1  
  
    assert(coroutine.resume(APP))
```

```
elseif (evt.key=="") and INPUT then
```

```
    EMAIL = EMAIL .. '@'
```

```
    redraw()
```

```
    assert(coroutine.resume(APP))
```

```
-- Deleta último caractere.
```

```
elseif (evt.key=='CURSOR_LEFT') and INPUT then
```

```
    INPUT.text = string.sub (INPUT.text, 1, -2)
```

```
        EMAIL = string.sub(EMAIL, 1, -2)

        assert(coroutine.resume(APP))

    else

        EMAIL = EMAIL .. evt.key

        assert(coroutine.resume(APP))

    end

end

end

end

end

-- Main.

require 'conecta'

APP = coroutine.create (

    function ()

        event.register (handler_keypress)

        INPUT = { text='|', font={ face='vera', height=22 },

            color='green', x=0, y=0, draw=draw_text }
```

```
-- Loop para digitação do e-mail

while (true) do

    INPUT.text = EMAIL..'!'

    DRAWS[#DRAWS + 1] = INPUT

    redraw ()

    -- Espera pela tecla entrada.

    coroutine.yield ()

    -- Remove o texto que estãoj em DRAWS.

    DRAWS[#DRAWS] = nil

    if enviar then

        MENSAGEM = { text='Obrigado por responder a nossa enquete!',
font={face='vera', height=14},

        color='green', x=0, y=0, draw=draw_text }

        MENSAGEM2 = { text='Sua resposta serÃ; enviada por e-mail em breve!',
font={face='vera', height=14},

        color='green', x=0, y=16, draw=draw_text }
```

```
DRAWS[#DRAWS + 1] = MENSAGEM

DRAWS[#DRAWS + 1] = MENSAGEM2

redraw ()

-- Usa a função "conectar" em "conecta.lua", enviando o e-mail e o voto por
parâmetro

conectar(EMAIL, 'respostamusica')

coroutine.yield ()

DRAWS[#DRAWS] = nil

DRAWS[#DRAWS] = nil

break

end

end

-- Evento que atribui a ação "stop" Ã propriedade "encerra", declarada no ncl
-- na mídia "Email_musica". Faz com que a mídia pare na execução do ncl.

local evt = {
```

```
        class = 'ncl',  
        type = 'presentation',  
        name = 'encerra',  
        action = 'stop',  
    }  
    event.post(evt)  
  
end)  
  
-- Registra evento handler  
  
event.register (  
    function (evt)  
        if evt.class == 'ncl' and evt.type == 'presentation' and evt.action == 'start' then  
            coroutine.resume (APP)  
        end  
    end  
end  
)  
  
-- Fim: Email_musica.lua
```



**5) Arquivo *tcp.lua***

```
-- $Id: tcp.lua,v 1.1 2009/02/13 15:42:10 root Exp $

--[[
-- TODO:
-- * não aceita `tcp.execute` reentrante
--]]

local _G, coroutine, event, assert, pairs, type
    = _G, coroutine, event, assert, pairs, type
local s_sub = string.sub

module 'tcp'

local CONNECTIONS = {}

local current = function ()
    return assert(CONNECTIONS[assert(coroutine.running())])
end

local resume = function (co, ...)
    assert(coroutine.status(co) == 'suspended')
    assert(coroutine.resume(co, ...))
    if coroutine.status(co) == 'dead' then
        CONNECTIONS[co] = nil
    end
end
```

end

function handler (evt)

if evt.class ~= 'tcp' then return end

if evt.type == 'connect' then

for co, t in pairs(CONNECTIONS) do

if (t.waiting == 'connect') and

(t.host == evt.host) and (t.port == evt.port) then

t.connection = evt.connection

t.waiting = nil

resume(co)

break

end

end

return

end

if evt.type == 'disconnect' then

for co, t in pairs(CONNECTIONS) do

if t.waiting and

(t.connection == evt.connection) then

t.waiting = nil

resume(co, nil, 'disconnected')

end

end

```
    return
end

if evt.type == 'data' then
    for co, t in pairs(CONNECTIONS) do
        if (t.waiting == 'data') and
            (t.connection == evt.connection) then
            resume(co, evt.value)
        end
    end
end

return
end

end

event.register(handler)

function execute (f, ...)
    resume(coroutine.create(f), ...)
end

function connect (host, port)
    local t = {
        host = host,
        port = port,
        waiting = 'connect'
    }
    CONNECTIONS[coroutine.running()] = t
end
```

```
event.post {  
    class = 'tcp',  
    type = 'connect',  
    host = host,  
    port = port,  
}  
  
return coroutine.yield()  
end  
  
function disconnect ()  
    local t = current()  
    event.post {  
        class = 'tcp',  
        type = 'disconnect',  
        connection = assert(t.connection),  
    }  
end  
  
function send (value)  
    local t = current()  
    event.post {  
        class = 'tcp',  
        type = 'data',  
        connection = assert(t.connection),
```

```
    value = value,  
  }  
end
```

```
function receive (pattern)
```

```
  pattern = pattern or " -- TODO: '*1'/number
```

```
  local t = current()
```

```
  t.waiting = 'data'
```

```
  t.pattern = pattern
```

```
  if s_sub(pattern, 1, 2) ~= '*a' then
```

```
    return coroutine.yield()
```

```
  end
```

```
  local all = "
```

```
  while true do
```

```
    local ret = coroutine.yield()
```

```
    if ret then
```

```
      all = all .. ret
```

```
    else
```

```
      return all
```

```
    end
```

```
  end
```

```
end
```

**APÊNDICE D – CÓDIGO FONTE ARQUIVOS PHP****1) Arquivo *respostaagronomia.php***

```
<?php  
  
$to    = $_REQUEST["email"];  
  
$subject = 'Enquete TV Digital - UNICEUB';  
  
$message = 'Ola,  
  
O UniCEUB agradece o seu interesse pelo curso de AGRONOMIA!  
  
Em breve enviaremos maiores informações a respeito.  
  
UniCEUB  
  
www.uniceub.br';  
  
$headers = 'From: faleconosco@uniceub.br' . "\r\n" .  
  
    'Reply-To: wanessa.bastos@gmail.com' . "\r\n" .  
  
mail($to, $subject, $message, $headers);  
  
/* echo "email: ". $to ; */  
  
?>
```

**2) Arquivo *respostamoda.php***

```
<?php
```

```
$to = $_REQUEST["email"];
```

```
$subject = 'Enquete TV Digital - UNICEUB';
```

```
$message = 'Ola,
```

```
O UniCEUB agradece o seu interesse pelo curso de MODA!
```

```
Em breve enviaremos maiores informações a respeito.
```

```
UniCEUB
```

```
www.uniceub.br';
```

```
$headers = 'From: faleconosco@uniceub.br' . "\r\n" .
```

```
    'Reply-To: wanessa.bastos@gmail.com' . "\r\n" .
```

```
mail($to, $subject, $message, $headers);
```

```
/* echo "email: ". $to ; */
```

```
?>
```

**3) Arquivo *respostamusica.php***

```
<?php
```

```
$to = $_REQUEST["email"];
```

```
$subject = 'Enquete TV Digital - UNICEUB';
```

```
$message = 'Ola,
```

```
O UniCEUB agradece o seu interesse pelo curso de MUSICA!
```

```
Em breve enviaremos maiores informações a respeito.
```

```
UniCEUB
```

```
www.uniceub.br';
```

```
$headers = 'From: faleconosco@uniceub.br' . "\r\n" .
```

```
    'Reply-To: wanessa.bastos@gmail.com' . "\r\n" .
```

```
mail($to, $subject, $message, $headers);
```

```
/* echo "email: ". $to ; */
```

```
?>
```