

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**PROTÓTIPO DE SISTEMA DE ACESSO A DISPOSITIVOS
ELETRÔNICOS ATRAVÉS DA TV DIGITAL INTERATIVA**

RODRIGO BRÜNING WESSLER

BLUMENAU
2009

2009/2-19

RODRIGO BRÜNING WESSLER

**PROTÓTIPO DE SISTEMA DE ACESSO A DISPOSITIVOS
ELETRÔNICOS ATRAVÉS DA TV DIGITAL INTERATIVA**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciência
da Computação — Bacharelado.

Prof. Francisco Adell Péricas, Mestre - Orientador

**BLUMENAU
2009**

2009/2-19

PROTÓTIPO DE SISTEMA DE ACESSO A DISPOSITIVOS ELETRÔNICOS ATRAVÉS DA TV DIGITAL INTERATIVA

Por

RODRIGO BRÜNING WESSLER

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Francisco Adell Péricas, Mestre – Orientador, FURB

Membro: _____
Prof. Oscar Dalfovo, Doutor – FURB

Membro: _____
Prof. Sérgio Stringari, Mestre – FURB

Blumenau, 17 de dezembro de 2009

Dedico este trabalho a minha namorada Bruna,
que esteve ao meu lado todo este tempo.

AGRADECIMENTOS

A Deus, por estar comigo me carregando durante as madrugadas dedicadas a este trabalho.

À minha família, pelo apoio, cuidado e carinho que me possibilitaram chegar até aqui.

À minha namorada, Bruna, pela paciência, incentivo, compreensão e amor durante todo o trabalho principalmente nos momentos mais difíceis.

Aos meus amigos, pela ajuda, incentivo e por me aturarem nas longas conversas sobre este trabalho.

À empresa Senior Sistemas, pela oportunidade de aprendizado e pelo entendimento das ausências.

À fundação CERTI, principalmente a Andrei pela oportunidade de visitar e conhecer um dos lugares onde a TV digital brasileira está nascendo.

Ao meu orientador, Francisco, por ter acreditado e instruído para conclusão deste trabalho.

Felizes os que dão risada das suas tolices, pois deles é a fonte do relaxamento.

Augusto Cury

RESUMO

A possibilidade de agregar poder computacional a um aparelho de TV traz significativas mudanças no dia-a-dia das pessoas, permitindo o acesso a várias ferramentas e conteúdos antes disponíveis apenas em computadores. A medida que os sistemas e tecnologias evoluem, torna-se cada vez mais necessária uma forma prática de controlar todos os dispositivos encontrados em uma residência, facilitando a execução de tarefas diárias. Baseado nestas premissas, este trabalho demonstra através do uso da TV digital as características e recursos de um aplicativo capaz de centralizar as informações de todos os dispositivos presentes em uma casa.

Palavras-chave: TV digital. Interatividade. Gíngã. Domótica.

ABSTRACT

The possibility to add computational power to a television bring significant changes to the day by day of the people, allowing the access to many tools and contents that before were available only for computers. The way that the systems and technologies evolve, every time become more necessary one practice way to control all the devices found in a house, making easier the execution of the daily tasks. Based in this premises, this work will demonstrate through the use of the digital television the features and resources of an application to centralize the information of all devices in a house.

Key-words: Digital television. Interactivity. Ginga. Domotics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de uma aplicação de governo eletrônico.....	15
Figura 2 – <i>Set-top box</i>	19
Figura 3 - <i>Set-top box</i>	20
Figura 4 - Painel traseiro de um <i>set-top box</i> para ligação a periféricos.....	22
Figura 5 – Arquitetura do ATSC	24
Figura 6 – Arquitetura do DVB.....	25
Figura 7 – Arquitetura do ISDB	27
Figura 8 – Acordo entre Brasil e Japão	29
Figura 9 – Arquitetura em alto nível do middleware Ginga.....	32
Figura 10 – Arquitetura Ginga-J e ambiente de execução.....	33
Figura 11 – Divisão das APIs no Ginga-J	34
Figura 12 – Contexto em que a pilha do software Ginga-J é executada	35
Figura 13 – Acesso através de um PDA	37
Figura 14 – Diagrama de casos de uso do protótipo	39
Quadro 1 – Detalhamento do caso de uso Cadastrar equipamento.....	40
Quadro 2 – Detalhamento do caso de uso Excluir equipamento.....	40
Quadro 3 – Detalhamento do caso de uso Selecionar equipamento	41
Quadro 4 – Detalhamento do caso de uso Selecionar função	41
Quadro 5 – Detalhamento do caso de uso Consultar detalhes de uma função ..	42
Quadro 6 – Detalhamento do caso de uso Executar função	42
Figura 15 – Diagrama principal de classes.....	43
Figura 16 – Diagrama de classes do pacote <code>view</code>	44
Figura 17 – Diagrama de classes do pacote <code>connection</code>	45
Figura 18 – Diagrama de classes do pacote <code>components</code>	46
Figura 19 – Diagrama de classe do pacote <code>reader</code>	47
Figura 20 – Diagrama de classes do sub-pacote <code>components</code>	48
Figura 21 - Diagrama de sequência do caso de uso Cadastrar equipamento	49
Figura 22 – Diagrama de sequência do caso de uso Executar função.....	50
Figura 23 – Tela inicial do XletView	53
Figura 24 – Console de execução do XletView	53

Figura 25 – Comando para ativar o <i>debug</i> remoto no XletView.....	54
Figura 26 – Exemplo de arquivo XML interpretado pela aplicação	54
Quadro 7 – Exemplo de rotina que desenha um componente	55
Quadro 8 – Parte do código responsável pela leitura do arquivo XML	55
Figura 27 – Exemplo de resposta do comando <code>getFunctions</code>	56
Figura 28 – Tela inicial do protótipo	57
Figura 29 – Menu principal do protótipo	58
Figura 30 – Tela de inclusão de dispositivos.....	59
Figura 31 – Tela que exclui um dispositivo	60
Figura 32 – Tela de seleção de dispositivos	61
Figura 33 – Tela de seleção de função	62
Figura 34 – exemplo de tela com as ações de uma função.....	63

LISTA DE SIGLAS

ABERT/SET – Associação Brasileira de Emissoras de Rádio e Televisão/Sociedade de Engenharia de Televisão

ABNT – Associação Brasileira de Normas Técnicas

API – *Application Programming Interface*

ARIB – *Association of Radio Industries and Businesses*

ARIB STD-B24 – *Data Coding and Transmission Specification for Digital Broadcasting*

ARIB-STD B23 – *Application Execution Engine Platform for Digital Broadcasting*

ATSC – *Advanced Television Systems Committee*

BML – *Broadcast Markup Language*

COM-TV – Comissão Assessora para Assuntos de Televisão

DASE – *DTV Application Software Environment*

DAVIC – *Digital Audio Video Council*

DOM – *Document Object Model*

DVB – *Digital Video Broadcasting*

FCC – *Federal Communications Commission*

GEM – *Globally Executable MHP*

Ginga-CC – Ginga Common-Core

GPL – *General Public License*

HAVI – *Home Audio Video Interoperability*

HDTV – *High Definition Television*

IDE – *Integrated Development Environment*

ISDB – *Integrated Services Digital Broadcasting*

JMF – *Java Media Framework*

MPEG – *Moving Picture Experts Group*

MPH – *Multimedia Home Platform*

NCL – *Nested Context Language*

PDA – *Personal Digital Assistant*

PVR – *Personal Video Recorder*

SBTVD – Sistema Brasileiro de TV Digital

TCP/IP – *Transmission Control Protocol/Internet Protocol*

UIT – União Internacional de Telecomunicações

UML – *Unified Modeling Language*

URD – Unidade Receptora decodificadora

USB – *Universal Serial Bus*

XML – *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 TV DIGITAL.....	17
2.2 TV DIGITAL INTERATIVA	18
2.3 RECEPTOR DIGITAL	19
2.3.1 Arquitetura	20
2.3.2 Interfaces de comunicação	21
2.4 PADRÕES MUNDIAIS DE TV DIGITAL.....	22
2.4.1 ATSC	23
2.4.1.1 DASE.....	24
2.4.2 DVB	25
2.4.2.1 MHP.....	26
2.4.3 ISDB.....	26
2.4.3.1 ARIB.....	27
2.5 SISTEMA BRASILEIRO DE TV DIGITAL.....	27
2.5.1 Ginga.....	30
2.5.1.1 Ginga Common-Core.....	31
2.5.1.2 Ginga-NCL	32
2.5.1.3 Ginga-J.....	33
2.6 DOMÓTICA.....	35
2.7 TRABALHOS CORRELATOS	36
3 DESENVOLVIMENTO	38
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	38
3.2 ESPECIFICAÇÃO	38
3.2.1 Diagrama de casos de uso	39
3.2.2 Diagrama de classes	42
3.2.3 Diagrama de sequência	48
3.3 IMPLEMENTAÇÃO	50
3.3.1 Técnicas e ferramentas utilizadas.....	50

3.3.1.1 Ambiente de desenvolvimento Eclipse.....	51
3.3.1.2 Implementação dos testes	51
3.3.1.3 Emulador XletView	51
3.3.1.4 Formulário XML	54
3.3.1.5 Protocolo de comunicação	56
3.3.2 Operacionalidade da implementação	56
3.4 RESULTADOS E DISCUSSÃO	63
4 CONCLUSÕES.....	65
4.1 EXTENSÕES	66
REFERÊNCIAS BIBLIOGRÁFICAS	67

1 INTRODUÇÃO

O advento da TV digital permitiu, além da qualidade de imagem e som superior, a possibilidade de interatividade e o oferecimento de novos serviços (MONTEZ, BECKER, 2005, p. 59).

Ainda segundo Montez e Becker (2005, p. 45), o conceito de interatividade nunca esteve tão em destaque, com programas interativos, comerciais interativos, sites, televisão. Tudo parece ficar mais moderno e atraente se possuir esta característica.

Segundo Brugnera (2006), domótica é um recurso utilizado para controle de um ou mais aparelhos eletrônicos em uma residência por meio de uma central computadorizada. O termo surgiu da junção da palavra “domus”, que significa residência, com as palavras eletrônica e informática. Os equipamentos e a tecnologia utilizados são os mesmos utilizados em ambientes industriais, com a diferença básica da finalidade e do comportamento.

A televisão é um dos mais importantes meios de difusão de informações e entretenimento para grande parte da humanidade. Por causa da televisão, lugares remotos são colocados em pé de igualdade com áreas urbanas altamente desenvolvidas (MINISTÉRIO DAS COMUNICAÇÕES, 2003).

Com a TV digital, os aparelhos de TV passam a possuir poder computacional semelhante ao dos computadores, transferindo para a TV um leque de aplicações e serviços quase tão abrangente quanto dos computadores. Conforme CETIC.BR (2007), 98% dos domicílios possuem televisão e 74 % dos lares têm telefone celular, enquanto apenas 24% possuem computadores de mesa.

Segundo Morris (2005, p. 20), do ponto de vista do consumidor, em um primeiro momento pode não haver uma mudança radical no que vêem entre a TV digital e transmissão analógica. As mudanças que o espectador é exposto são evolutivas ao invés de revolucionárias. A TV digital oferece quatro vantagens principais:

- a) uma quantidade maior de canais;
- b) melhor qualidade de imagem;
- c) maior qualidade de som;
- d) os serviços e aplicativos adicionais.

Segundo Ministério das Comunicações (2003), a TV Digital poderá ainda incluir aplicações totalmente desvinculadas da programação dita normal, relacionadas ao trabalho ou negócios, ao entretenimento, à educação e à informação, bem como aplicações baseadas no

protocolo IP.

Um exemplo de uma aplicação eletrônica (T-Govern) é apresentada na Figura 1.



Fonte: Soares e Barbosa (2006).

Figura 1 – Exemplo de uma aplicação de governo eletrônico

Mesmo no contexto global, atualmente poucos sistemas aproveitam os recursos disponíveis na tecnologia da TV digital para integrar a televisão com outros equipamentos de uma residência.

Diante do exposto, este trabalho estuda e pesquisa o Sistema Brasileiro de TV Digital (SBTVD) e desenvolve um sistema capaz de controlar equipamentos ligados a uma rede local, possibilitando a execução de algumas funções deles através do controle remoto da TV, disponibilizando assim mais uma alternativa de comodidade ao dia-a-dia das pessoas.

1.1 OBJETIVOS DO TRABALHO

O Objetivo deste trabalho é desenvolver um sistema capaz de reconhecer e interagir com dispositivos eletrônicos residenciais através da TV digital interativa.

Os objetivos específicos do trabalho são:

- a) identificar equipamentos residenciais, como por exemplo um computador, ligados a uma rede local utilizando *Transmission Control Protocol/Internet Protocol* (TCP/IP);
- b) propor um protocolo para a comunicação destes dispositivos eletrônicos com a TV digital;
- c) controlar funcionalidades destes equipamentos, como por exemplo o envio de mensagens de texto, através do controle remoto da TV.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos. No capítulo 2 é descrita a fundamentação teórica utilizada como referência para este trabalho.

O capítulo 3 é descreve o desenvolvimento do trabalho e a especificação do protótipo.

Por fim, o capítulo 4 descreve as conclusões obtidas com o trabalho juntamente com algumas sugestões para realização de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos relacionados a TV digital, nos quais este trabalho está fundamentado.

2.1 TV DIGITAL

A Televisão surge a partir de uma série de estudos em vários países. O invento da TV, no início, é caracterizado pelo uso de um sistema mecânico, idealizado pelo Alemão Paul Nipkow em 1884. A luz passa por um disco perfurado que faz o esquadramento da imagem, gerando um fluxo de elétrons que são transmitidos por um fio (CROCOMO, 2007, p. 48).

Ainda segundo Crocomo (2007, p. 60), aos poucos os componentes digitais passaram a ser incorporados aos aparelhos de televisão, como por exemplo, o controle remoto. Os sintonizadores dos aparelhos de TV passaram a ser digitais, possibilitando a troca de canais sem a necessidade de virar os botões. Este pode ser considerado o primeiro aprendizado no uso de recursos digitais.

Segundo Ministério das Comunicações (2003), a TV digital não deve ser vista apenas como uma evolução tecnológica da televisão. Trata-se de uma nova plataforma de comunicação baseada em tecnologia digital para a transmissão de sinais.

Grande parte das emissoras brasileiras já trabalha com recursos digitais nas gravações de externas e em estúdio, edição no computador (não-linear), bem como nas mesas onde são selecionados os sons e imagens que vão ao ar. A grande mudança para que todo o processo se complete é que a transmissão passe a ser digital e a recepção nas casas também (CROCOMO, 2007, p. 60).

Com base nessa visão histórica, é possível notar que, também em busca do mercado doméstico, no caso para a venda de aparelhos de som e imagem, recursos, antes só disponíveis no setor profissional, acabam incorporados em aparelhos domésticos. (CROCOMO, 2007, p. 52).

Os equipamentos digitais melhoraram muito em relação aos analógicos e permitem um padrão de produção melhor. Outra contribuição é a compressão dos arquivos de vídeo, ou

seja, a criação de arquivos pequenos com qualidade aceitável para transmissão via internet. Os vídeos podem ser recebidos pela internet e transmitidos pela TV que garante uma interatividade real e mais ampla (CROCOMO, 2007, p. 53).

2.2 TV DIGITAL INTERATIVA

Segundo Steuer (1992), interatividade é relacionada a extensão de quanto um usuário pode participar ou influenciar na modificação imediata, na forma e no conteúdo de um ambiente computacional.

Lemos (1997) compreende interatividade como uma forma de interação técnica, de características eletrônico-digital, e que se diferencia da interação analógica que caracteriza a mídia tradicional.

Segundo Montez e Becker (2005, p. 54), o conceito de TV interativa não está claro em nenhum lugar do mundo, suscitando inúmeras controvérsias. Mas uma característica é praticamente unânime, a TV deixa de ser unidirecional.

Ainda conforme Montez e Becker (2005, p. 55), é possível classificar toda a variedade de informações incorporadas pelo termo em sete grandes grupos:

- a) TV avançada: a qualidade de som e vídeo aumentam, passando a possuir resolução 16:9, igual a de cinema;
- b) internet TV: permite acesso a internet e todas as suas funções usando o aparelho de TV;
- c) TV individualizada: permite escolher ângulos de câmeras, repetição de cenas perdidas, organização das janelas, etc;
- d) vídeo sob demanda: os programas podem ser assistidos a qualquer hora. Pode ser comparado ao acesso a uma vídeo locadora;
- e) *Personal Video Recorder* (PVR): permite a gravação digital de programas;
- f) *walled garden*: um portal contendo um guia de aplicações interativas;
- g) console de jogos: permite o uso da TV para jogos, usando como adversário a própria TV ou em rede, contra outros jogadores.

2.3 RECEPTOR DIGITAL

A TV digital necessita de um receptor que pode estar embutido em uma televisão digital ou ser um equipamento a parte (Figura 2), o que é mais comum devido ao preço de uma TV totalmente digital e também pela convergência sob demanda para a tecnologia digital. Neste caso o receptor passa a ser chamado de *set-top box* ou Unidade Receptora decodificadora (URD).

Segundo Montez e Becker (2005, p. 106), para permitir ao telespectador a navegação na internet, ou a interação com serviços, os *set top boxes* possuem capacidade de processamento, contendo um hardware com tecnologias comuns ao mundo da computação.

É sobre este hardware que o *middleware* da TV digital é executado.



Fonte: Guimarães (2005, p. 18).

Figura 2 – *Set-top box*

Na Figura 3 pode-se observar um *set-top box* desenvolvido para o padrão brasileiro de TV digital.



Fonte: Fundação CERTI (2008).

Figura 3 - *Set-top box*

2.3.1 Arquitetura

O *set-top box* possui uma arquitetura semelhante a de um computador comum (SILVA, 2003, p. 39):

- a) *System Board*: é por aqui que passam todas as informações referentes à televisão digital, esta informação é compartilhada pelos demais componentes;
- b) sintonizadores: trabalham com a recepção dos sinais das redes digitais baseadas nas modulações existentes;
- c) modulador e demodulador: nesta etapa é verificada a existências de possíveis erros e depois passado para o demultiplexador;
- d) demultiplexador e *decrytor*: são circuitos integrados que identificam os pacotes com formato de dados particulares como vídeo, áudio ou serviços interativos.

Também são responsáveis por descriptografar informações recebidas criptografadas e depois enviar para o decodificador;

- e) decodificadores: separam para converter os bits recebidos num formato que pode ser ouvido e visto, exemplo disso é um decodificador de vídeo que transforma os pacotes de vídeo recebidos numa seqüência de imagens, formatando estas para diferentes tipos de resoluções. Depois que os dados foram interpretados são enviados para o processador do equipamento;
- f) CPU: é a da parte mais importante do *set-top box*, já que é onde se encontra o *chip* do processador. Suas principais funções são: inicialização dos vários componentes da *set-top box*, processamento de aplicações da Internet e da TV interativa, monitorização e administração das interrupções de hardware, retirada de dados e interrupções da memória, execução de vários programas. Os principais *chips* da CPU pertencem às seguintes famílias: ARM, MIPS, PowerPC, SparcRISC, STx0, SH-4 Series, X86;
- g) configuração de memória: utilizada para armazenar e manipular informações, e também para trabalhar com vídeos que exigem uma maior resolução;
- h) recursos de armazenamento: na primeira geração dos *Set-top boxes* existia apenas uma memória volátil chamada *flash*, porém hoje já há *winchesters* ligados com uma grande capacidade de armazenamento. Nestes discos pode-se armazenar documentos, e-mails, além disso, podem ser usados para gravações de vídeos digitais e outras informações.

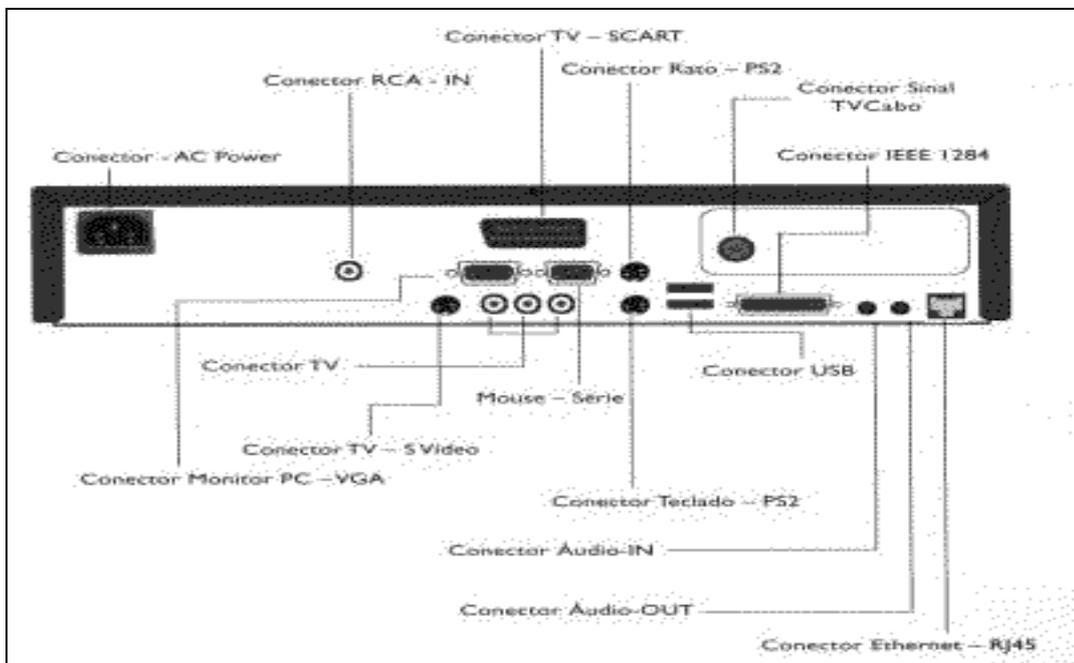
2.3.2 Interfaces de comunicação

Da mesma forma como em um computador, os *set-top boxes* podem fazer uso de várias interfaces para comunicação com o mundo exterior (Figura 4), existindo a possibilidade de comunicar o *set-top box* com aparelhos como: câmera de vídeo, DVD, teclado, etc.

Em Sun Microsystems (2001) são citados alguns exemplos de interfaces, entre as quais estão:

- a) IEEE-1284 (Interface Paralela) – utilizada para conexão a impressoras;
- b) RS-232 (Interface Serial) – utilizada para conectar impressoras, PCs e modems;
- c) *Universal Serial Bus* (USB) – utilizada para conectar teclados, *joysticks*, mouses e vários outros equipamentos ao *set-top box*;

- d) IEEE-1394 (*Firewire*) – conexão de alta velocidade para filmadoras e outros dispositivos áudios-visuais;
- e) 10 Base-T (*Ethernet*) – conexão em rede local com PCs e impressoras;
- f) PCMCIA tipo II – vários usos como, por exemplo, estender a capacidade de memória do *set-top-box*, deciframento de acesso condicional para serviços de várias redes diferentes, aumentar a capacidade de armazenamento e adicionar novos sintonizadores ao *set-top box*;
- g) *SmartCards* – permite acesso a diversos serviços de várias redes, além do uso em comércio eletrônico;
- h) controles remotos e teclados sem fio.



Fonte: Silva (2003, p. 41).

Figura 4 - Painel traseiro de um *set-top box* para ligação a periféricos

2.4 PADRÕES MUNDIAIS DE TV DIGITAL

No mundo hoje existem três padrões principais de televisão digital:

- a) ATSC - *Advanced Television Systems Committee*;
- b) DVB - *Digital Video Broadcasting*;
- c) ISDB - *Integrated Services Digital Broadcasting*.

Segundo Curtiss (2000 apud CROCOMO, 2007 p. 61), os três padrões foram testados no Brasil no período de outubro de 1999 a abril de 2000, e a transmissão foi feita da TV Cultura de SP (Fundação Padre Anchieta) para pontos distantes até 40 quilômetros. Como, no Brasil, a grande maioria da população tem acesso a canais abertos de TV, a análise levou em consideração a sintonia por esse sistema (terrestre), sendo que, na época, o sistema japonês foi o que teve melhores resultados, seguido pelo europeu.

A seguir é apresentado uma breve descrição sobre cada um dos três principais padrões de TV digital e suas características.

2.4.1 ATSC

O *Advanced Television Systems Committee* (ATSC) foi desenvolvido nos Estados Unidos por um grupo de empresas, entre elas a AT&T, GI (Chicago'S General Instrument Corp.), MIT (Massachusetts Institute of Technology), Philips Electronics North America Corporation, David Sarnoff Research Center, Thompson Consumer Electronics e Zenith Electronics Corporation (SILVA, 2003, p. 28).

Ainda segundo Silva (2003, p. 28), este padrão foi adotado pela *Federal Communications Commission* (FCC) na década de 90 e também é utilizado na Argentina, Taiwan, Coréia do Sul e no Canadá, onde foi adotado o padrão digital em 1997 pela empresa Canadá DTV Inc.

Segundo Coelho (2005, p. 24), o interesse principal em seu desenvolvimento era a transmissão de serviços de TV em alta definição, mais conhecido como *High Definition Television* (HDTV). O HDTV permite uma maior definição de vídeo e fidelidade no áudio, chegando a 6-1 canais de distribuição do som.

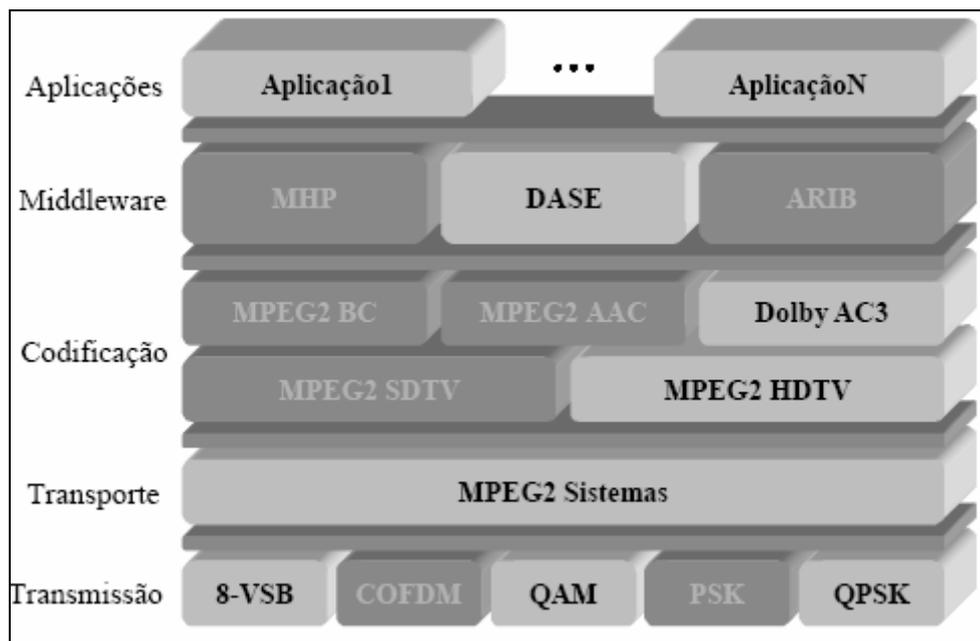
Segundo Anatel (2001 apud MONTEZ, BECKER, 2005, p. 120) este padrão, utiliza modulação 8-VSB, possuindo uma taxa de transmissão de 19,8 Mbps, ocupando uma largura de banda de 6,7 ou 8MHz. O ATSC apresenta problemas na recepção por antenas internas e não permite a recepção móvel.

Conforme Montez e Becker (2005, p. 120), a multiplexação e a codificação de vídeo são feitas sobre o padrão MPEG-2. Já a codificação de vídeo é realizada através do padrão *Dolby AC-3* (Figura 5).

No Brasil testes realizados pela Associação Brasileira de Emissoras de Rádio e Televisão/Sociedade de Engenharia de Televisão (ABERT/SET) mostram que o padrão

ATSC apresenta problemas em distâncias grandes com recebimento do sinal de 80%, enquanto o ideal seria 100% (SILVA, 2003, p. 29).

Ainda segundo Silva (2003, p. 29) outro problema deste padrão é causado pelo efeito de refração, ou seja, as ondas transmitidas se refletem em um objeto próximo em movimento e retornam criando interferências. Como o sinal é digital, ou tem sinal ou não tem, logo estas interferências podem retirar o sinal do ar. A ABERT/SET realizou testes com transmissões próximas a estações ferroviárias, e toda vez que passava um trem havia o corte do sinal.



Fonte: Coelho (2005, p. 4).

Figura 5 – Arquitetura do ATSC

2.4.1.1 DASE

O *DTV Application Software Environment* (DASE) foi desenvolvido pelo ATSC como um padrão norte-americano para a camada de *middleware* em *set-top boxes* de TVs digitais. (MONTEZ, BECKER, 2005, p. 120).

Ainda segundo Montez e Becker (2005, p. 120), o DASE adota uma máquina virtual Java como mecanismo que facilita a execução de aplicações interativas. Também permite o uso de linguagens declarativas, usadas na *web*, como HTML e JavaScript.

2.4.2 DVB

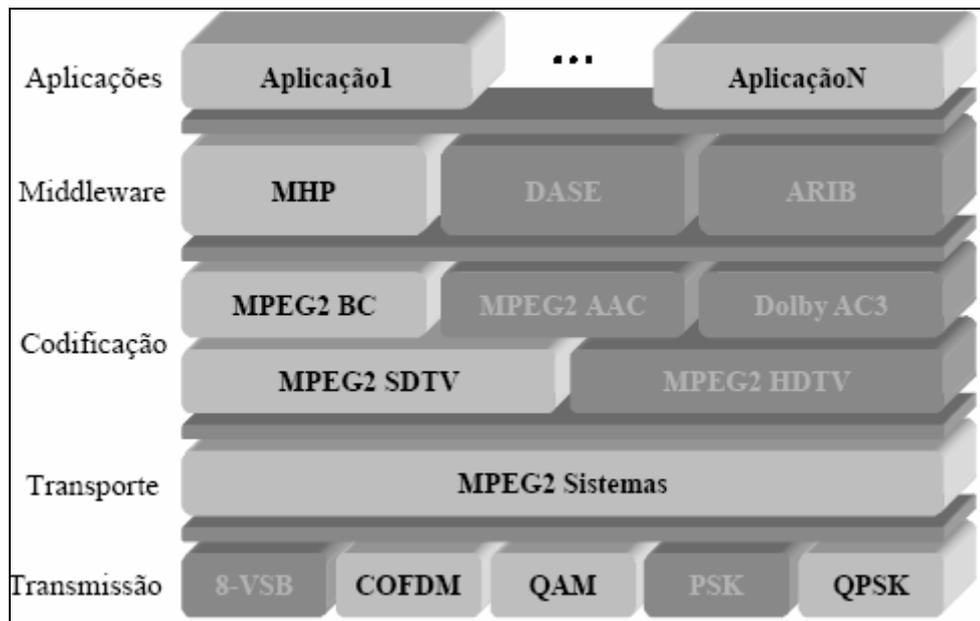
O *Digital Video Broadcasting* (DVB) é conhecido como o sistema europeu de televisão digital. Na realidade este padrão é formado por um conjunto de documentos, definindo padrões de transmissão, sendo os mais conhecidos (MONTEZ, BECKER, 2005, p.116,):

- a) DVB-T: radiodifusão;
- b) DVB-C: difusão por cabo;
- c) DVB-S: difusão por satélite;
- d) DVB-MPH: (*Multimedia Home Plataform*): padrão de *middleware*.

Conforme Montez e Becker (2005, p. 117), este conjunto de padrões é definido por um consórcio homônimo, que começou oficialmente em setembro de 1993. O consórcio DVB atualmente é composto por mais de trezentos membros, de 35 países.

Ainda segundo Montez e Becker (2005, p. 117), o padrão DVB é adotado em países da Europa, além da Austrália, Malásia, Hong Kong, Índia, África do Sul, entre outros. O país mais avançado na consolidação do uso do DVB é a Inglaterra, já possuindo mais de um milhão de usuários.

Na Figura 6 é demonstrado como é implementada a arquitetura do DVB.



Fonte: Coelho (2005, p.5).

Figura 6 – Arquitetura do DVB.

2.4.2.1 MHP

O *Multimedia Home Platform* (MHP) é o *middleware* desenvolvido para o padrão DVB, começou a ser especificado em 1997 com sua primeira versão lançada em junho de 2000 (MONTEZ, BECKER, 2005, p. 118).

Ainda segundo Montez e Becker (2005, p. 118), seu ambiente é baseado no uso de uma máquina virtual Java e um conjunto de Interfaces de Programação de Sistemas (*Application Programming Interface* – API). Essas APIs possibilitam aos programas escritos em Java o acesso a recursos e facilidades do receptor digital de forma padronizada.

Além do uso da API Java, a partir da versão 1.1 o MHP introduziu a possibilidade de usar uma linguagem semelhante ao HTML denominada DVB-HTML. Estas aplicações assim como as feitas usando as APIs Java possuem a capacidade de (MONTEZ, BECKER, 2005, p. 118):

- a) *download*, através de um canal de interatividade, aplicações interativas;
- b) armazenar aplicações em memória persistentes;
- c) acessar leitores de *smart cards*;
- d) controlar aplicações de internet, tais como navegador *web* e leitor de *e-mail*.

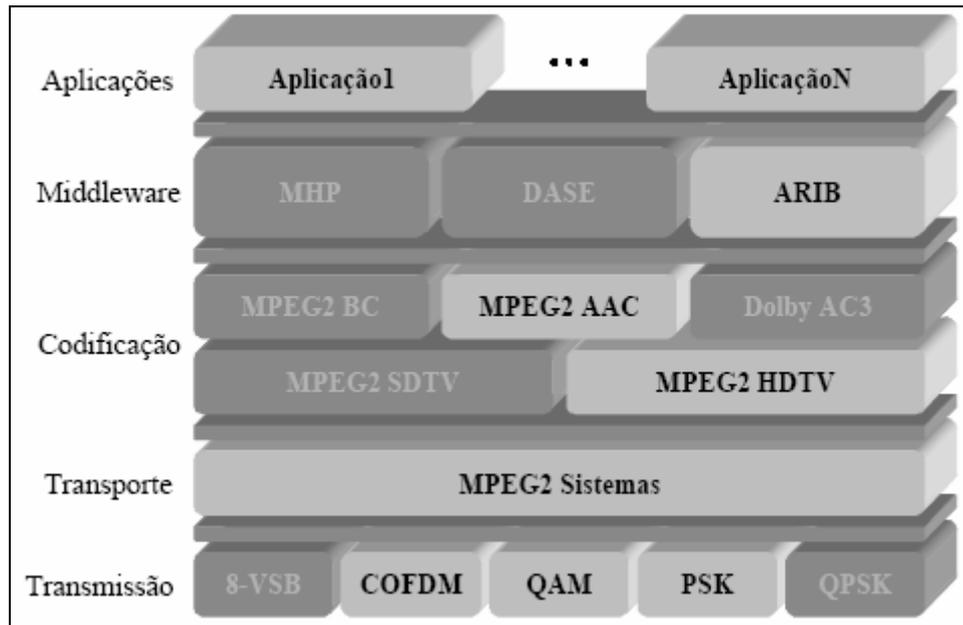
Em janeiro de 2003 foi publicada a especificação do *Globally Executable MHP* (GEM) que é a tentativa de harmonizar diferentes *middleware* em um padrão único.

2.4.3 ISDB

Criado em 1999 por várias empresas e operadoras de televisão, o *Integrated Services Digital Broadcasting* (ISDB) é o sistema de transmissão japonês, sendo adotado somente pelo próprio Japão (MONTEZ, BECKER, 2005, p. 121).

O ISDB utiliza a modulação Cofdm, com algumas variações, possui uma taxa de transferência que varia entre 3,5 a 23,23 Mbps, e uma largura de banda de 6,7 ou 8MHz (MONTEZ, BECKER, 2005, p. 121).

Ainda conforme Montez e Becker (2005, p. 121), as suas maiores vantagens são a grande flexibilidade de operação e o potencial para transmissões móveis e portáteis. A multiplexação e a codificação de vídeo, são realizadas em MPEG-2. A codificação de áudio utiliza o MPEG-2 ACC (Figura 7).



Fonte: Coelho (2005, p. 6).

Figura 7 – Arquitetura do ISDB

2.4.3.1 ARIB

O *Association of Radio Industries and Businesses* (ARIB) é o *middleware* do ISDB.

Este *middleware* é formado por alguns padrões, como o *Data Coding and Transmission Specification for Digital Broadcasting* (ARIB STD-B24), que defini uma linguagem declarativa denominada *Broadcast Markup Language* (BML). Esta linguagem, é baseada na linguagem padrão de serviços web *Extensible Markup Language* (XML), sendo usada na especificação de serviços multimídia para a TV digital (MONTEZ, BECKER, 2005, p. 122).

Ainda segundo Montez e Becker (2005, p. 122) outra especificação é o *Application Execution Engine Platform for Digital Broadcasting* (ARIB-STD B23), que é baseada na especificação DVB-MHP, e traduz uma tendência do ARIB de tentar estabelecer um núcleo comum entre o seu padrão de *middleware*, o MHP e o DASE.

2.5 SISTEMA BRASILEIRO DE TV DIGITAL

Segundo Freitas (2004, p. 6), a criação do Grupo Técnico ABERT/SET de TV Digital

em 1994 marcou o início dos estudos acerca do tema no País. As concessionárias de radiodifusão integrantes do Grupo tinham a intenção de acompanhar a evolução da tecnologia e de propor soluções ao governo, representado então pela Comissão Assessora para Assuntos de Televisão (COM-TV) do Ministério das Comunicações.

Segundo Ministério das Comunicações (2003), para o Brasil, um padrão de televisão digital deve contemplar as reais necessidades da sociedade brasileira, tendo em vista o perfil de renda da população e as novas possibilidades abertas através da interatividade.

As principais premissas adotadas são:

- a) estabelecer e aumentar a rede de competências nacional, promovendo a efetiva integração das pesquisas brasileiras nas áreas de abrangência desse programa;
- b) apresentar solução técnica inovadora, mantendo e aproveitando a compatibilidade com elementos já padronizados no mercado mundial de TV Digital;
- c) ser flexível às condições sócio-econômicas do Brasil;
- d) aproveitar o parque nacional instalado de televisores;
- e) permitir uma implantação gradual, minimizando os riscos e os custos para a sociedade, procurando soluções escaláveis e evolutivas, minimizando legados;
- f) ser configurável para potencial adoção por outros países, facilitando exportação;
- g) promover o adensamento da cadeia de valor e de geração de negócios baseados no sistema de TV Digital, consolidando os atores envolvidos;
- h) promover soluções industriais que favoreçam a economia de escala;
- i) promover a cultura digital com a TV interativa.

Ainda segundo Ministério das Comunicações (2003), antes da escolha de desenvolver um modelo Brasileiro de TV digital todos os três padrões mundiais de TV foram amplamente testados e analisados por entidades e especialistas brasileiros.

Conforme Montez e Becker (2005, p. 134) o relatório final dos testes de TV digital confirmou o melhor desempenho do padrão japonês, considerado superior devido ao melhor desempenho na recepção de sinais televisivos em ambientes fechados, e a sua flexibilidade para recepção de programas ou acesso a serviços através de terminais fixos ou móveis.

Conforme HXD (2009), em 29 de junho de 2006, o governo brasileiro publicou o Decreto Presidencial Nº 5.820, onde defini a escolha do padrão japonês como base para o padrão brasileiro.

Segundo Crocomo (2007, p. 65) na cerimônia (Figura 8) foi anunciado também que, além de optar pela transmissão com uso do padrão japonês de modulação, foram adotados componentes criados no Brasil, como *middleware* Ginga, terminais de acesso de baixo custo e

o método de compressão de dados H.264, diferente do utilizado no Japão. Um sistema que o governo chamou de nipo-brasileiro.



Fonte: HDX (2006).

Figura 8 – Acordo entre Brasil e Japão

Conforme Ministério das Comunicações (2003), este programa de desenvolvimento tecnológico deverá trazer, entre outros, os seguintes benefícios para o Brasil:

- a) soluções voltadas para uma sociedade bastante desigual (buscando promover inclusão digital e social);
- b) fortalecimento das redes de competência e aumento da base de conhecimento (engenharia de sistemas);
- c) redução da dependência tecnológica e substituição de importações de softwares e componentes;
- d) incentivo à produção de conteúdos regionais e locais e novas oportunidades de negócio (geração de empregos);
- e) inserção efetiva da C&T brasileira nos consórcios internacionais;
- f) maior capacidade de articulação e poder de negociação com fornecedores internacionais (redução do pagamento de *royalties*);
- g) potencial de exportação de softwares e equipamentos;
- h) fortalecimento da indústria nacional com produção de alta escala;
- i) novos mecanismos de suporte à cultura, educação e saúde.

Segundo DTV (2009), o padrão brasileiro adotado pelo Peru, Argentina, Chile e Venezuela é reconhecido pelo braço da ONU regulador de telecomunicações a União Internacional de Telecomunicações (UIT), que passa a recomendar as normas brasileiras de radiodifusão digital.

Ainda segundo DTV (2009), o Brasil passa oficialmente a exportador de tecnologia de

TV digital com o mesmo status dos EUA e Europa. As normas da Associação Brasileira de Normas Técnicas (ABNT) foram publicadas no portal da UIT, traduzidas nas línguas oficiais do órgão: inglês, árabe, chinês, espanhol, francês e russo.

Conforme HDX (2009), o SBTVD possui sete normas técnicas, são elas:

- a) transmissão - especifica o padrão de transmissão do sistema brasileiro de televisão digital terrestre, compreendendo o sistema de codificação de canal e modulação e descrevendo o processamento de sinal no modulador e os processos de demodulação na recepção;
- b) codificação - especifica a multiplexação de sinais para radiodifusão digital (áudio, vídeo e dados) dos mecanismos de transporte e da estrutura de dados aplicáveis ao sistema brasileiro de televisão digital terrestre;
- c) multiplexação - especifica as tabelas básicas de informação de serviço, conhecidas por tabelas SI, para os sinais de radiodifusão que fazem parte da transmissão de dados do sistema brasileiro de televisão digital terrestre;
- d) receptores - especifica o conjunto de funcionalidades essenciais requeridas aos terminais de acesso destinados à recepção fixa, móvel e portátil;
- e) gestão de direitos - especifica os mecanismos de proteção contra cópia através de proteções nas interfaces de saída de vídeo, áudio e dados, assim como o protocolo de regras de controle de cópias e exibições aplicáveis ao sistema brasileiro de televisão digital terrestre;
- f) *middleware* - especifica o modelo de referência que possibilita a difusão de dados que integra o sistema de difusão digital definido como Sistema Brasileiro de Televisão Digital;
- g) canal de interatividade - as especificações contidas nesse documento formam o conjunto de funcionalidades essenciais requeridas aos receptores de sinais de televisão digital assim como dos dispositivos externos para o canal de interatividade.

2.5.1 Ginga

Conforme Ginga (2009), Ginga é o nome do *middleware* do Sistema Brasileiro de TV Digital.

Segundo o Portal do Software Público Brasileiro (2009a), GINGA é a camada de

software intermediário (*middleware*) que permite o desenvolvimento de aplicações interativas para a TV digital de forma independente da plataforma de hardware dos fabricantes de terminais de acesso (*set-top boxes*).

O Ginga pode ser dividido em três subsistemas principais: Ginga-CC, Ginga-J e Ginga-NCL.

2.5.1.1 Ginga Common-Core

Segundo Associação Brasileira de Normas Técnicas (2007), o Ginga Common-Core (Ginga-CC) oferece o suporte básico para os ambientes de programação declarativo e procedural. Conforme o Portal do Software Público Brasileiro (2009b), o Ginga-CC tem como principais funções a exibição dos vários objetos de mídia, o controle do plano gráfico, o tratamento de dados obtidos do carrossel¹ de objetos, o tratamento do canal de retorno, entre outras.

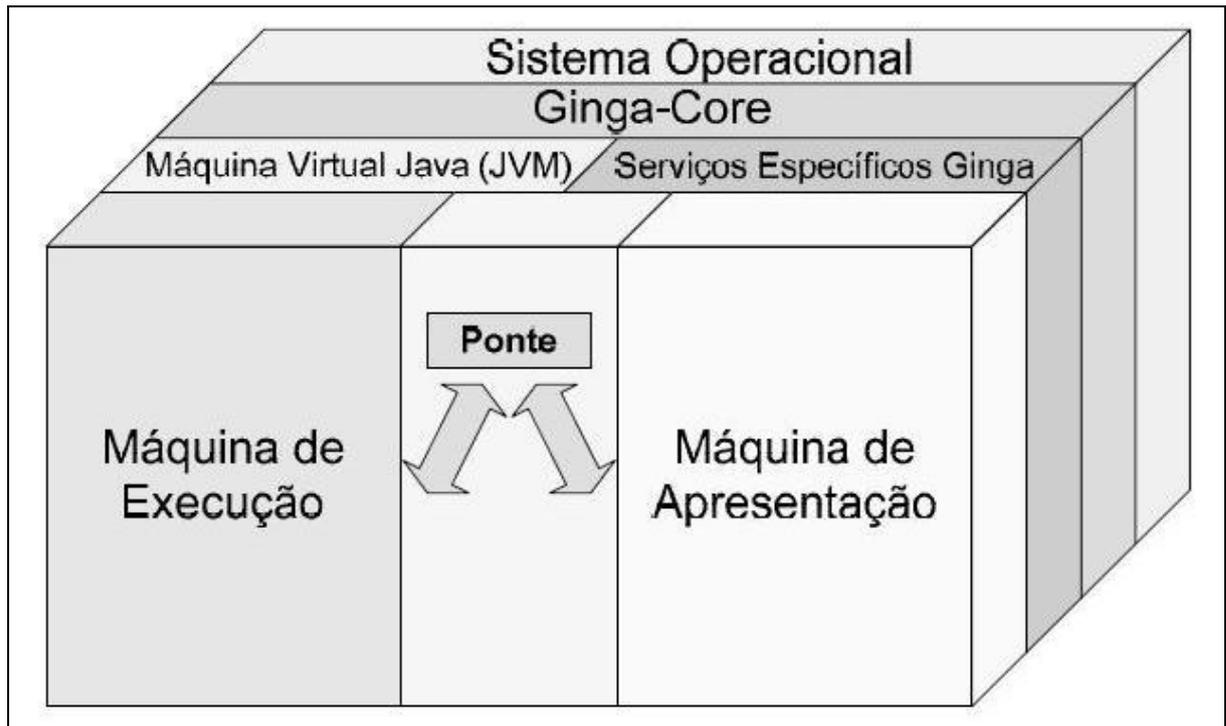
Um conteúdo declarativo deve ser baseado em uma linguagem declarativa, isso é, em uma linguagem que enfatiza a descrição declarativa do problema, ao invés da sua decomposição em uma implementação algorítmica.

Um conteúdo procedural deve ser baseado em uma linguagem não declarativa. Linguagens não declarativas podem seguir diferentes paradigmas. Tem-se assim, as linguagens baseadas em módulos, orientadas a objetos entre outras.

Ainda segundo Associação Brasileira de Normas Técnicas (2007), numa programação procedural, o computador deve obrigatoriamente ser informado sobre cada passo a ser executado. Como existem mais recursos disponíveis, o grau de complexidade é maior. A linguagem mais usual encontrada nos ambientes procedurais de um sistema de TV digital é Java.

O Portal do Software Público Brasileiro (2009b) afirma que uma aplicação não precisa ser puramente declarativa ou puramente procedural. Nos sistemas de TV digital, os dois tipos de aplicação coexistirão (Figura 9).

¹ Carrossel é um mecanismo de transmissão cíclica de objetos.



Fonte: Associação Brasileira de Normas Técnicas (2007).

Figura 9 – Arquitetura em alto nível do middleware Ginga

Ainda, segundo Portal do Software Público Brasileiro (2009b), 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 *Central Processing Unit* (CPU) quanto de ocupação de memória. Ao contrário dos outros sistemas, Ginga, desde seu projeto inicial, foi desenvolvido tendo em mente os dois ambientes de programação.

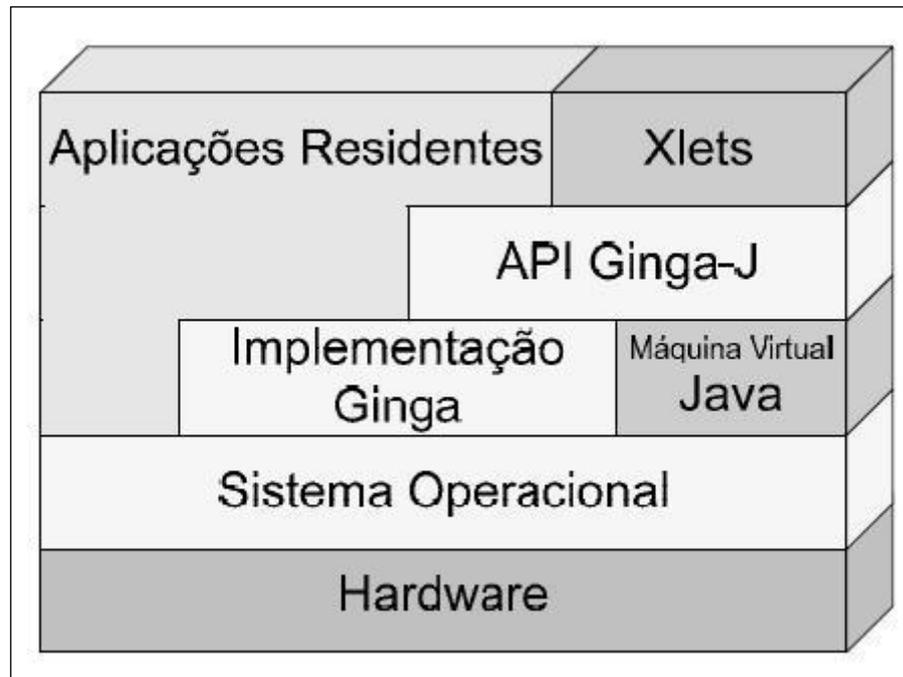
2.5.1.2 Ginga-NCL

Ginga-NCL foi desenvolvido pela PUC-Rio para prover uma infraestrutura de apresentação de aplicações baseadas em documentos hipermídia escritos na linguagem *Nested Context Language* (NCL), com facilidades para a especificação de aspectos de interatividade, sincronismo espaço temporal de objetos de mídia, adaptabilidade e suporte a múltiplos dispositivos. NCL possui Lua como sua linguagem de *script*.

2.5.1.3 Ginga-J

Ginga-J foi desenvolvido pela Universidade Federal da Paraíba (UFPB) para prover uma infraestrutura de execução de aplicações procedurais baseadas na linguagem Java.

O modelo Ginga-J distingue entre as entidades e recursos de hardware, software do sistema e aplicativos (Figura 10).



Fonte: Associação Brasileira de Normas Técnicas (2007).

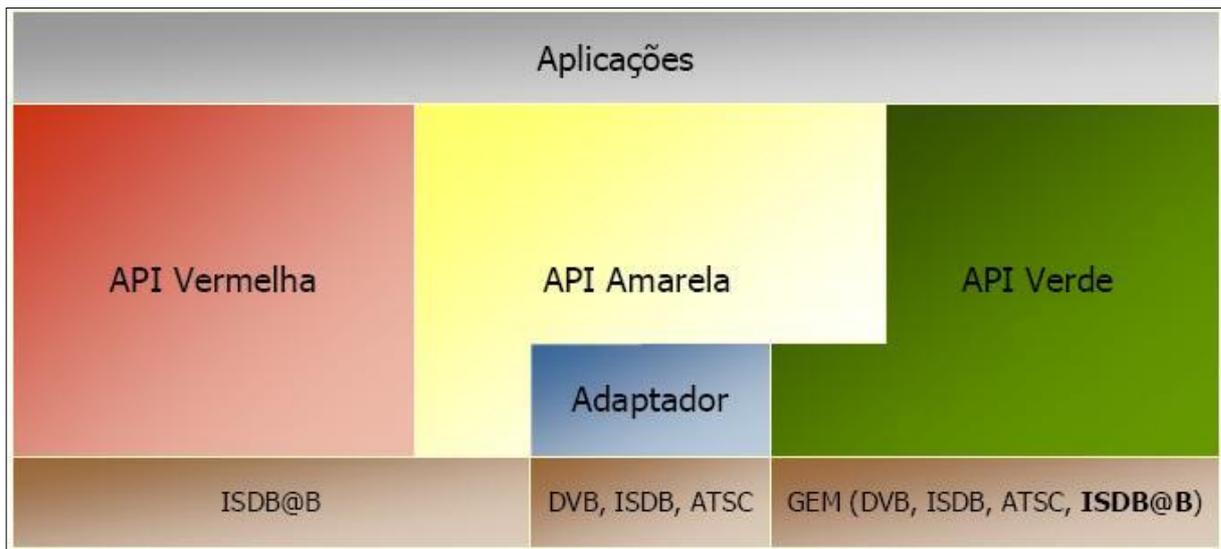
Figura 10 – Arquitetura Ginga-J e ambiente de execução

As aplicações residentes podem ser implementadas usando funções não padronizadas, fornecidas pelo Sistema Operacional (SO) do dispositivo de Ginga, ou por uma implementação particular do Ginga. Os aplicativos residentes também podem incorporar funcionalidades providas pelas APIs padronizadas Ginga-J. Aplicativos transmitindo Xlets² sempre devem utilizar API padronizadas fornecidas pelo Ginga-J.

Conforme Associação Brasileira de Normas Técnicas (2007), a definição Ginga-J é composta por API projetadas para suprir todas as funcionalidades necessárias para a implementação de aplicativos para televisão digital, desde a manipulação de dados multimídia até protocolos de acesso.

Segundo Portal do Software Público Brasileiro (2009a), o Ginga-J é dividido em três partes (Figura 11):

- a) APIs vermelhas: inovações que dão suporte às aplicações brasileiras, em especial as de inclusão social;
- b) APIs amarelas: também inovações brasileiras, mas que podem ser exportadas para os outros sistemas;
- c) APIs verdes: que seguem o núcleo comum do padrão GEM.



Fonte: Portal do Software Público Brasileiro (2009b).

Figura 11 – Divisão das APIs no Ginga-J

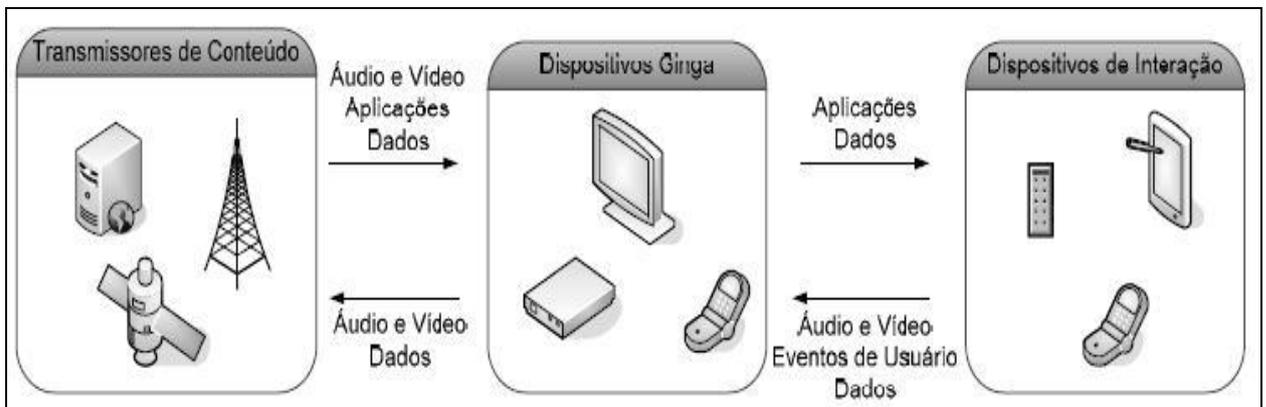
O Ginga-J é uma especificação de *middleware* distribuído, que reside em um dispositivo Ginga (dispositivo que embarque o *middleware* Ginga – um receptor de televisão digital), com possibilidade de possuir componentes de software nos dispositivos de interação (celulares, computadores, etc).

O telespectador pode interagir com o dispositivo Ginga através de dispositivos de interação que podem conter componentes do software, de forma que o dispositivo possa enviar informações para o Ginga utilizando as funcionalidades providas na especificação. Ou seja, estes componentes de software que podem ser instalados nos dispositivos de interação, permitem que as funcionalidades dos mesmos sejam exploradas, utilizando funcionalidades da API Ginga-J.

Por exemplo, um dispositivo de interação pode ser um *Personal Digital Assistant* (PDA) conectado à plataforma Ginga através de uma rede sem fio. Utilizando tal dispositivo de interação, um telespectador pode enviar comandos (eventos de usuário) à plataforma através do teclado do PDA e os aplicativos da plataforma podem enviar conteúdo visual para

² Xlet é a denominação de um programa feito para TV digital usando a API Java TV.

ser apresentado na tela do PDA (Figura 12).



Fonte: Associação Brasileira de Normas Técnicas (2007).

Figura 12 – Contexto em que a pilha do software Ginga-J é executada

2.6 DOMÓTICA

O termo domótica, resulta da junção do *latin domus* (casa) com a palavra robótica (automação). Foi adotado na Europa para designar o campo de aplicação tecnológica que visa a integração do espaço arquitetônico, da informática e das telecomunicações (ANGEL, 1993, p. 13).

Segundo Venturi (2005), a automação residencial é um novo domínio de aplicação tecnológica, tendo como objetivo básico melhorar a qualidade de vida, reduzindo o trabalho doméstico, aumentando o bem estar e a segurança de seus habitantes, visando também uma utilização racional e planejada dos diversos meios de consumo, procurando uma melhor integração através da automatização em áreas distintas como segurança, comunicação, controle e gestão de fluídos. Segundo as necessidades humanas, pode-se dividir a automação residencial em três partes distintas:

- a) necessidades de segurança:
 - proteção e vigilância contra intrusos,
 - prevenção de acidentes,
 - fugas de água ou gás;
- b) necessidades do ambiente:
 - térmico,

- acústico,
 - visual;
- c) necessidades de conforto nas atividades do dia-a-dia:
- dormir,
 - alimentar-se,
 - comunicar-se.

Segundo Brugnera (2006), conforme essa tecnologia vai ganhando espaço no mercado, esses equipamentos de controle vem sendo aprimorados cada vez com mais funções embutidas, como acesso à internet e conexões de redes.

Ainda segundo Brugnera (2006), utilizando sistemas de rede, como Internet e celulares, usuários podem ter acesso a todo o sistema, permitindo, por exemplo, controlar de fora equipamentos internos da casa, imagens de câmeras de segurança, e ainda fazer transferências de documentos armazenados em um banco de dados.

Devido ao elevado custo dos equipamentos necessários para que se possa implementar um tipo de sistema “inteligente” dentro de uma residência, a Domótica é pouco conhecida até mesmo para adeptos a área da informática. Mais um motivo: é a novidade. Com o aprimoramento desses equipamentos necessários pode diminuir o preço e se tornarem mais acessíveis. Provavelmente em um futuro não muito distante será parte da maioria das residências (BRUGNERA, 2006).

2.7 TRABALHOS CORRELATOS

Como trabalhos correlatos são descritos um trabalho acadêmico e uma ferramenta comercial que possui funcionalidades que serão desenvolvidas neste projeto.

Santos Neto (2005) apresenta um estudo sobre as tecnologias de comunicação inter-aplicações e uma análise das tecnologias desta natureza disponíveis em um ambiente de software de TV Digital. Ele propõe alguns cenários de referência para a validação das propostas e estudos realizados no trabalho, abordando a comunicação de uma aplicação no *set top box* usando a tecnologia de *web services*, como a comunicação celular – *set top box e set top box - celular*.

O Mordomus (MORDOMUS, 2008) é um sistema de gestão e controle de habitações inteligentes. É constituído pela integração do sistema físico de hardware, controlado pelo

software numa central de controle, possibilitando acesso local ou remoto a várias funções e configurações da casa, permitindo a gestão e o controle de qualquer equipamento conectado ao sistema: aquecedores, torradeira, microondas, máquinas de lavar, forno elétrico, câmeras de segurança, entre outros (Figura 13).



Fonte: Mordomus (2008).

Figura 13 – Acesso através de um PDA

3 DESENVOLVIMENTO

As sessões seguintes descrevem os requisitos, a especificação, a implementação e a operacionalidade do protótipo desenvolvido. Ao final deste capítulo, são indicados os resultados obtidos com este trabalho.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Na sequência são apresentados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF).

O sistema deverá:

- a) estar acessível através do menu da TV (RF);
- b) permitir cadastrar um equipamento conectado a rede local (RF);
- c) permitir excluir um equipamento cadastrado (RF).
- d) exibir todas as funções disponíveis de um equipamento no menu da TV (RF);
- e) permitir selecionar e executar uma função (RF);
- f) ser implementado usando a análise orientada a objetos (RNF);
- g) utilizar linguagem de programação Java (RNF);
- h) ser implementado utilizando o ambiente de desenvolvimento Eclipse (RNF);
- i) ser testado utilizando o ambiente de testes JUnit (RNF).

3.2 ESPECIFICAÇÃO

Na sequência é apresentada a especificação do protótipo, que foi modelada na ferramenta *Enterprise Architect* através da *Unified Modeling Language* (UML). Foram utilizados conceitos da orientação a objetos e *design patterns* para a criação dos diagramas de casos de uso, classe e de sequência. A seguir, são apresentados detalhadamente cada um desses documentos.

3.2.1 Diagrama de casos de uso

O protótipo possui seis casos de uso, que correspondem às suas principais funcionalidades: cadastrar equipamento, excluir equipamento, selecionar equipamento, selecionar função, consultar detalhes de uma função e executar função. Os seis casos de uso estão ilustrados na Figura 14.

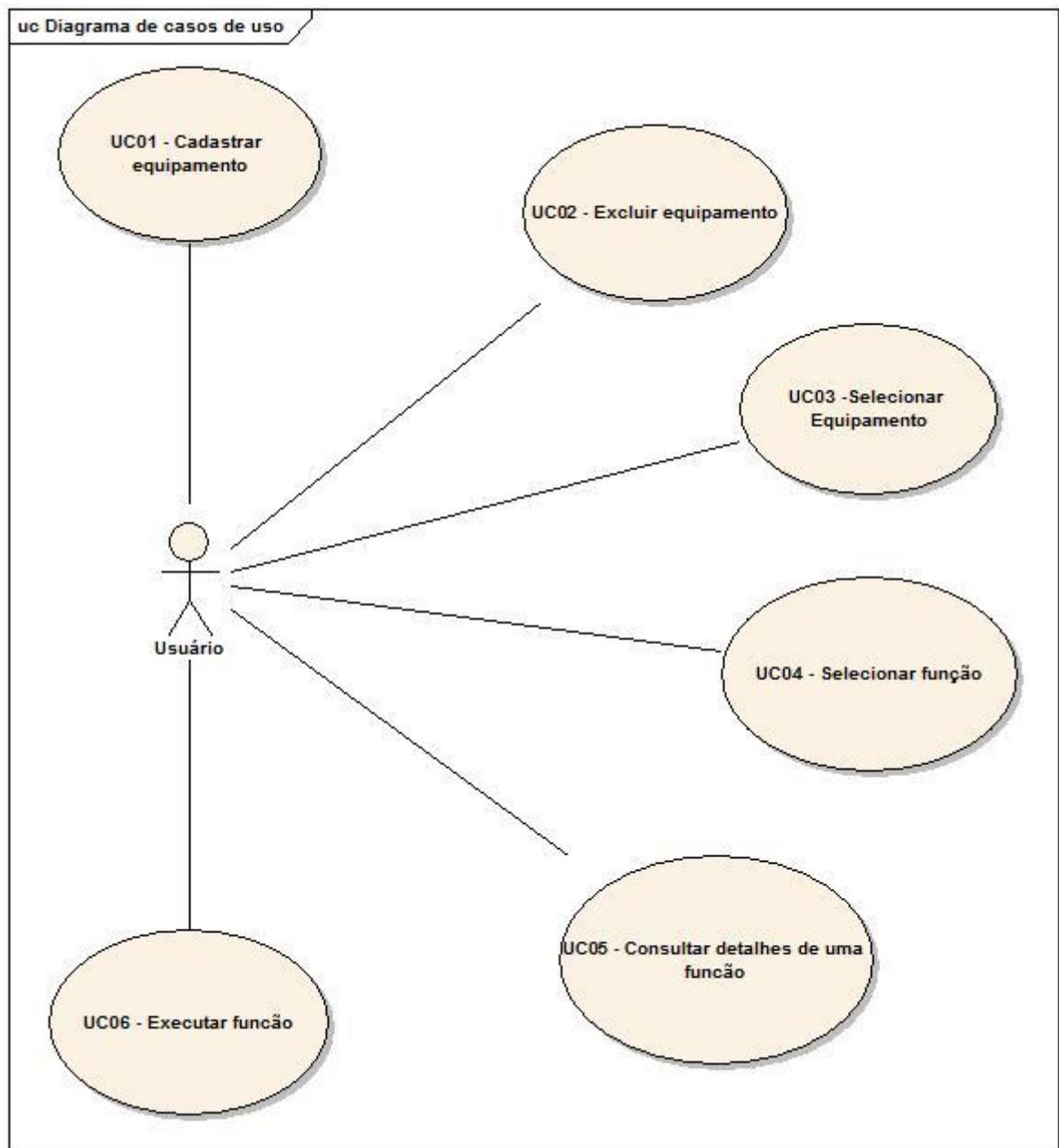


Figura 14 – Diagrama de casos de uso do protótipo

O primeiro caso de uso (Quadro 1), designado cadastrar equipamento, é o caso que apresenta quais são as ações realizadas pelo usuário para cadastrar um equipamento na aplicação.

UC01 – Cadastrar equipamento: permite ao usuário cadastrar um equipamento na aplicação.	
Pré-condições	Estar conectado a mesma rede do equipamento a ser adicionado.
Cenário principal	01) O usuário clica no botão “Incluir dispositivo”. 02) A aplicação solicita o preenchimento dos dados. 03) O usuário preenche os dados e confirma. 04) A aplicação valida os dados. 05) A aplicação grava o novo equipamento.
Fluxo alternativo 01	Não possui.
Exceção 01	No passo 04, caso os dados solicitados não tenham sido preenchidos, é apresentada ao usuário uma mensagem solicitando que os dados sejam preenchidos.
Pós-condições	O equipamento foi criado.

Quadro 1 – Detalhamento do caso de uso *Cadastrar equipamento*

O segundo caso de uso (Quadro 2), designado excluir equipamento, é o caso que apresenta quais são as ações realizadas pelo usuário para excluir um equipamento cadastrado na aplicação.

UC02 – Excluir equipamento: permite ao usuário excluir um equipamento cadastrado na aplicação.	
Pré-condições	Possuir equipamentos cadastrados na aplicação.
Cenário principal	01) O usuário clica no botão “Excluir dispositivo”. 02) A aplicação exibe uma lista com os equipamentos cadastrados. 03) O usuário seleciona um equipamento para exclusão. 04) O usuário clica no botão excluir ou no botão vermelho do controle remoto. 05) A aplicação valida exclusão. 06) A aplicação exclui o equipamento.
Fluxo alternativo 01	No passo 05, o usuário escolhe um equipamento protegido com senha. 05.01) A aplicação solicita senha. 05.02) O usuário informa senha.
Exceção 01	No passo 05.02, caso senha informada esteja incorreta, é apresentada ao usuário uma mensagem solicitando que a senha seja preenchida novamente.
Pós-condições	O equipamento foi excluído.

Quadro 2 – Detalhamento do caso de uso *Excluir equipamento*

O terceiro caso de uso (Quadro 3), designado selecionar equipamento, é o caso que apresenta quais são as ações realizadas pelo usuário para selecionar um equipamento cadastrado na aplicação.

UC03 – Selecionar equipamento: permite ao usuário selecionar um equipamento cadastrado na aplicação.	
Pré-condições	Possuir equipamentos cadastrados na aplicação.
Cenário principal	01) O usuário clica no botão “Selecionar dispositivo”. 02) A aplicação exibe uma lista com os equipamentos cadastrados. 03) O usuário seleciona um equipamento. 04) O usuário clica no botão “Selecionar” ou no botão verde do controle remoto. 05) A aplicação seleciona o equipamento.
Fluxo alternativo 01	No passo 04, o usuário escolhe um equipamento protegido com senha. 04.01) A aplicação solicita senha. 04.02) O usuário informa senha.
Exceção 01	No passo 04.02, caso senha informada esteja incorreta, é apresentada ao usuário uma mensagem solicitando que a senha seja preenchida novamente.
Pós-condições	O equipamento foi selecionado. Uma lista de funções é exibida.

Quadro 3 – Detalhamento do caso de uso Selecionar equipamento

O quarto caso de uso (Quadro 4), designado selecionar equipamento, é o caso que apresenta quais são as ações realizadas pelo usuário para selecionar um equipamento cadastrado na aplicação.

UC04 – Selecionar função: permite ao usuário selecionar uma função de um equipamento cadastrado na aplicação.	
Pré-condições	Possuir um equipamento selecionado.
Cenário principal	01) O usuário seleciona uma função da lista de funções. 02) O usuário clica no botão “Selecionar” ou no botão verde do controle remoto. 03) A aplicação seleciona a função.
Fluxo alternativo 01	No passo 02, o usuário escolhe um equipamento protegido com senha. 02.01) A aplicação solicita senha. 02.02) O usuário informa senha.
Fluxo alternativo 02	No passo 02, o usuário clica no botão amarelo do controle remoto. 02.01) A aplicação exibe uma descrição sobre função selecionada. 02.01) O usuário clica no botão azul para fechar a descrição.
Exceção 01	No passo 02, caso senha informada esteja incorreta, é apresentada ao usuário uma mensagem solicitando que a senha seja preenchida novamente.
Pós-condições	A função foi selecionada.

Quadro 4 – Detalhamento do caso de uso Selecionar função

O quinto caso de uso (Quadro 5), designado consultar detalhamento de uma função, é o caso que apresenta quais são as ações realizadas pelo usuário para exibir uma descrição sobre uma função.

UC05 – Consultar detalhes de uma função: permite ao usuário consultar detalhes de uma função.	
Pré-condições	Possuir uma função selecionada.
Cenário principal	01) O usuário clica no botão amarelo do controle remoto. 02) A aplicação exibe uma descrição sobre função selecionada. 03) O usuário clica no botão azul para fechar a descrição.
Fluxo alternativo 01	Não possui.
Exceção 01	Não possui.
Pós-condições	A descrição foi exibida para o usuário.

Quadro 5 – Detalhamento do caso de uso Consultar detalhes de uma função

O sexto caso de uso (Quadro 6), designado executar função, é o caso que apresenta quais são as ações realizadas pelo usuário para executar uma funcionalidade disponibilizada por um equipamento cadastrado.

UC06 – Executar função: permite ao usuário executar uma função selecionada.	
Pré-condições	Possuir uma função selecionada.
Cenário principal	01) O usuário clica no botão “Selecionar” ou no botão vermelho do controle remoto. 02) A aplicação exibe uma tela dinâmica de acordo com a função selecionada. 03) O usuário utiliza o formulário exibido. 04) O usuário clica no botão “Voltar” ou no botão azul para retornar ao menu principal da aplicação.
Fluxo alternativo 01	Não possui.
Exceção 01	Não possui.
Pós-condições	A funcionalidade foi utilizada.

Quadro 6 – Detalhamento do caso de uso Executar função

3.2.2 Diagrama de classes

O diagrama de classes fornece uma visão de como as classes definidas para desenvolver a aplicação estão estruturadas e relacionadas. De forma a facilitar o entendimento, são apresentados seis diagramas de classes.

O primeiro diagrama (Figura 15) fornece uma visão geral da aplicação, com as principais classes e interfaces de cada pacote.

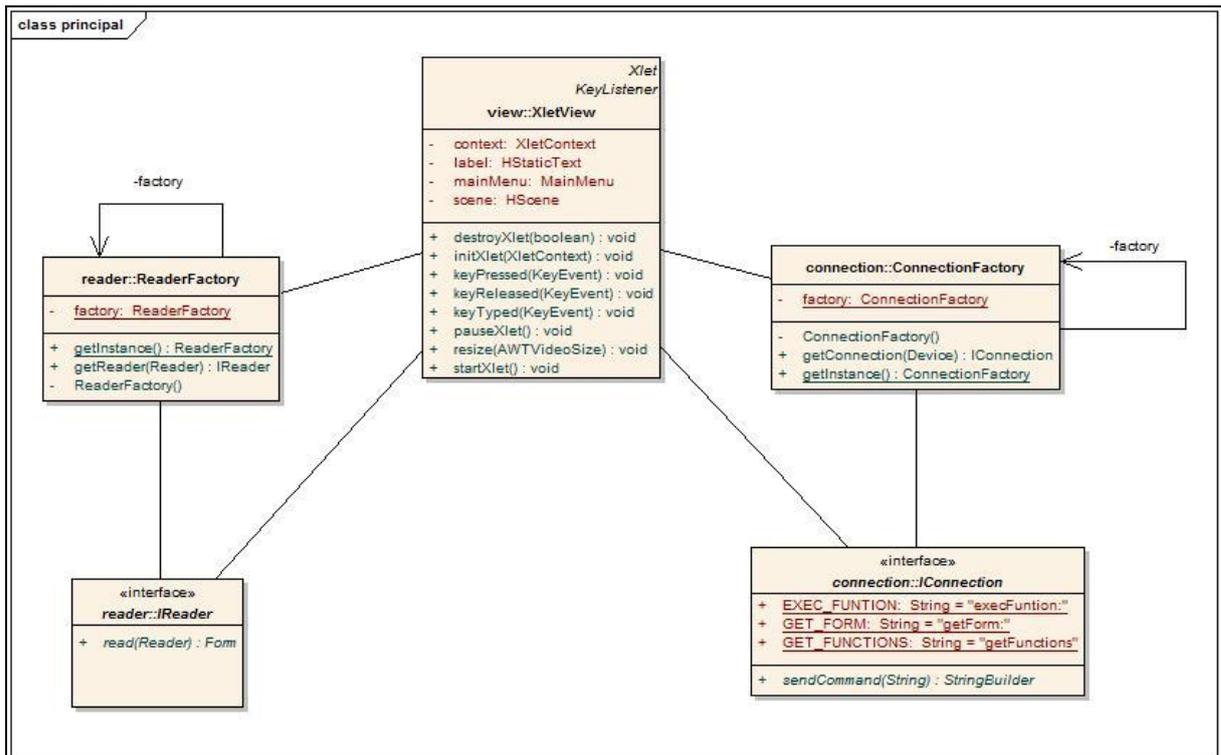


Figura 15 – Diagrama principal de classes

O pacote `view` contém as classes responsáveis pela interação com o usuário (Figura 16).

A principal classe deste pacote é a `XletView`, onde é implementada a `Xlet` e os recursos de controle de estado da execução aplicativo. Esta é a classe inicial da aplicação.

Para a tela inicial da aplicação foi implementada a classe `MenuPrincipal`, assim como as classes `DeviceRegister`, `DeviceRevome` e `DeviceSelect` para as três telas que são acessíveis a partir do menu principal.

A classe `DynamicRender` é responsável por gerar as telas dinâmicas que são recebidas dos dispositivos conectados a aplicação.

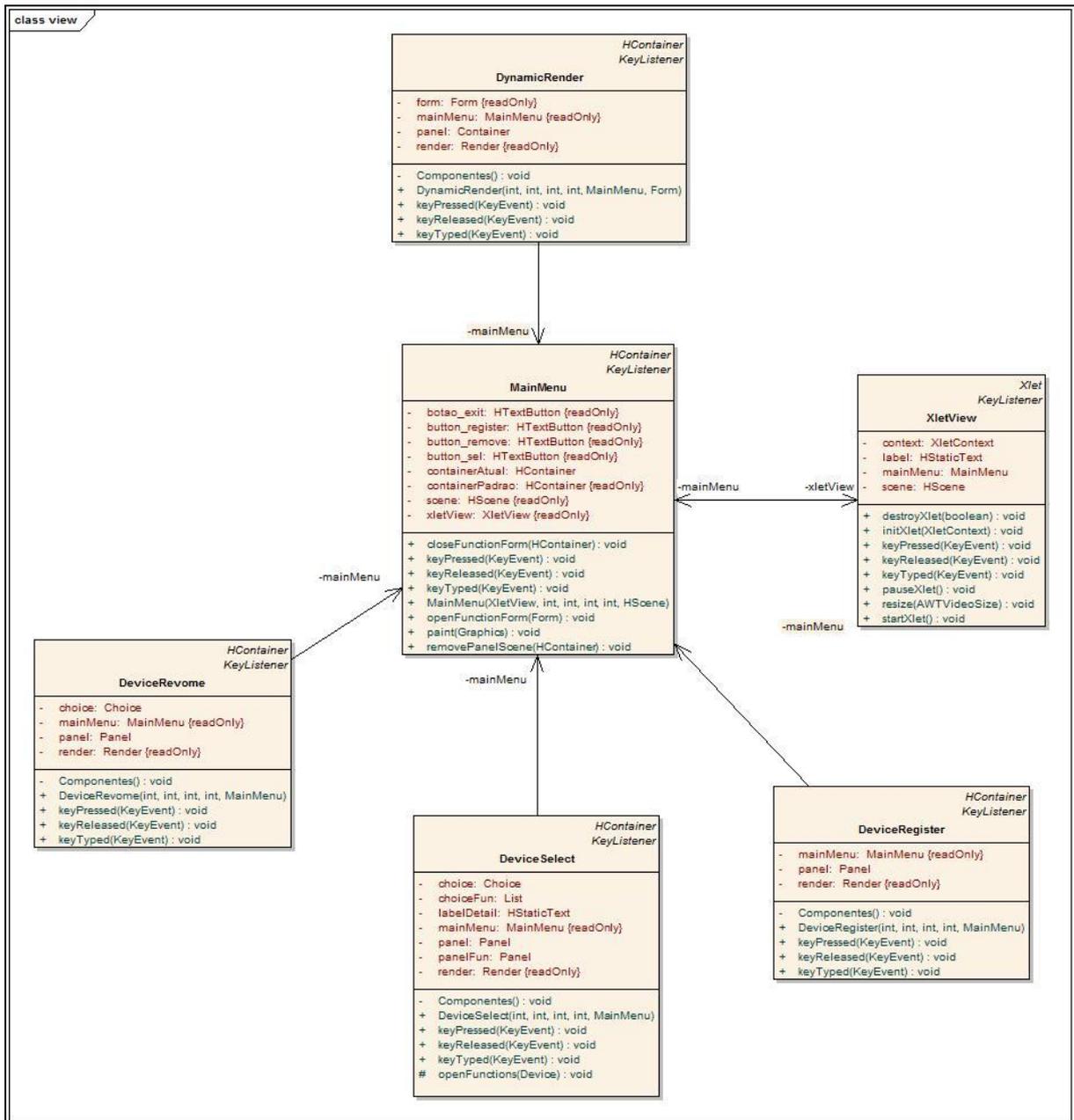


Figura 16 – Diagrama de classes do pacote view

O pacote `connection` contém as classes responsáveis pela conexão da aplicação com os demais dispositivos conectados a rede (Figura 17).

A principal classe deste pacote é a `TCPIPConnection`, que implementa a interface `IConnection` e controla os recursos e acesso a rede utilizando TCP/IP.

A classe `ConnectionFactory` é responsável criar uma classe que implemente a interface `IConnection` de acordo com o tipo de conexão desejado.

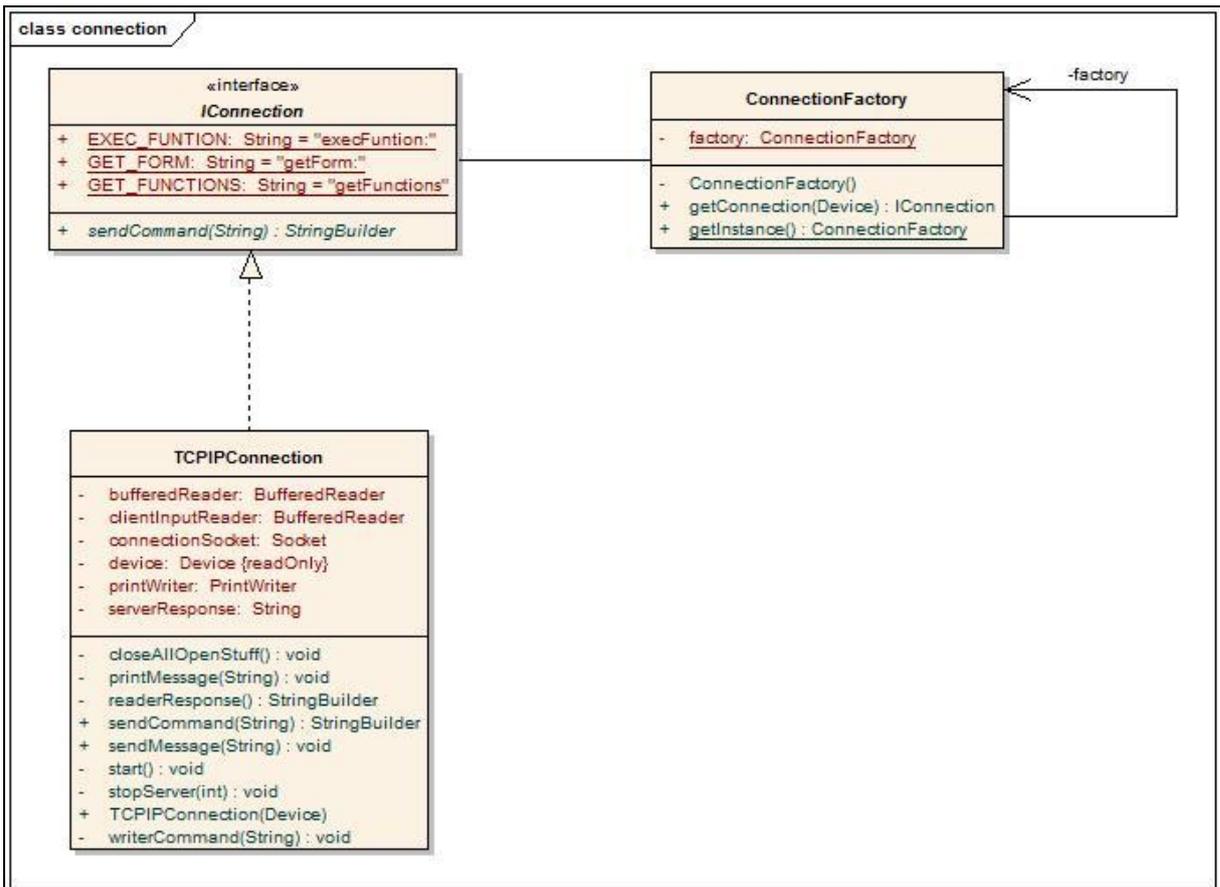


Figura 17 – Diagrama de classes do pacote connection

O pacote `components` contém as classes responsáveis pela estrutura dinâmica dos formulários (Figura 18).

A principal classe deste pacote é a `AbstractComponent`, que é a classe base de todos os componentes implementados nesta aplicação, esta classe possui os atributos e métodos comuns a todos os componentes.

Alguns exemplos de classes que representam componentes são as classes `Form`, `Button`, `Edit` e `Label`.

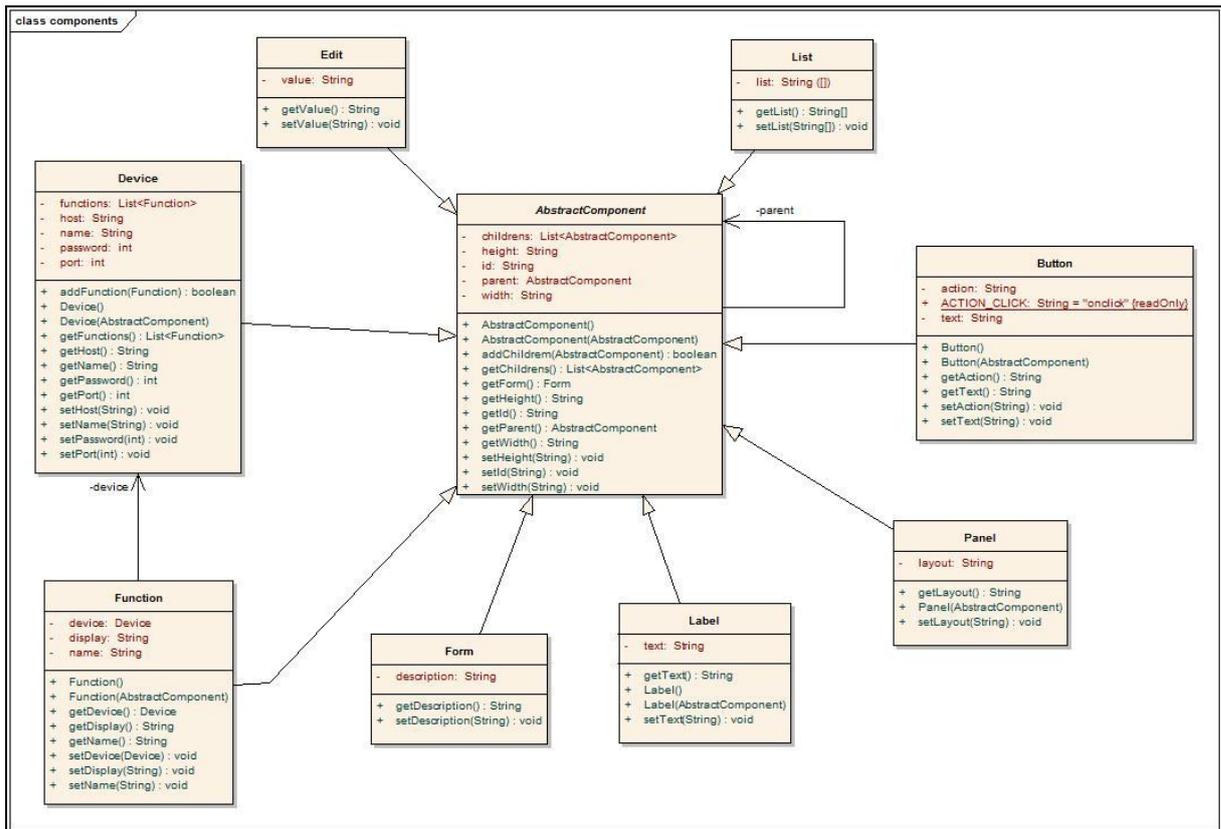


Figura 18 – Diagrama de classes do pacote components

O pacote `reader` e seus sub-pacotes `xml` e `components` contém as classes responsáveis pela leitura da definição de um formulário enviado por um dispositivo conectado a aplicação (Figura 19).

A principal classe do pacote `reader` é a `ReaderFactory`, que é responsável por criar uma classe que implemente a interface `IReader` de acordo com o tipo de leitura desejado.

No sub-pacote `xml` é que ficam as classes responsáveis pela leitura de formulários cujo a definição foi enviada em uma arquivo XML. Este pacote possui três classes `XMLContext`, `XMLHelper` e a classe principal `XMLReader` que implementa a interface `IReader`.

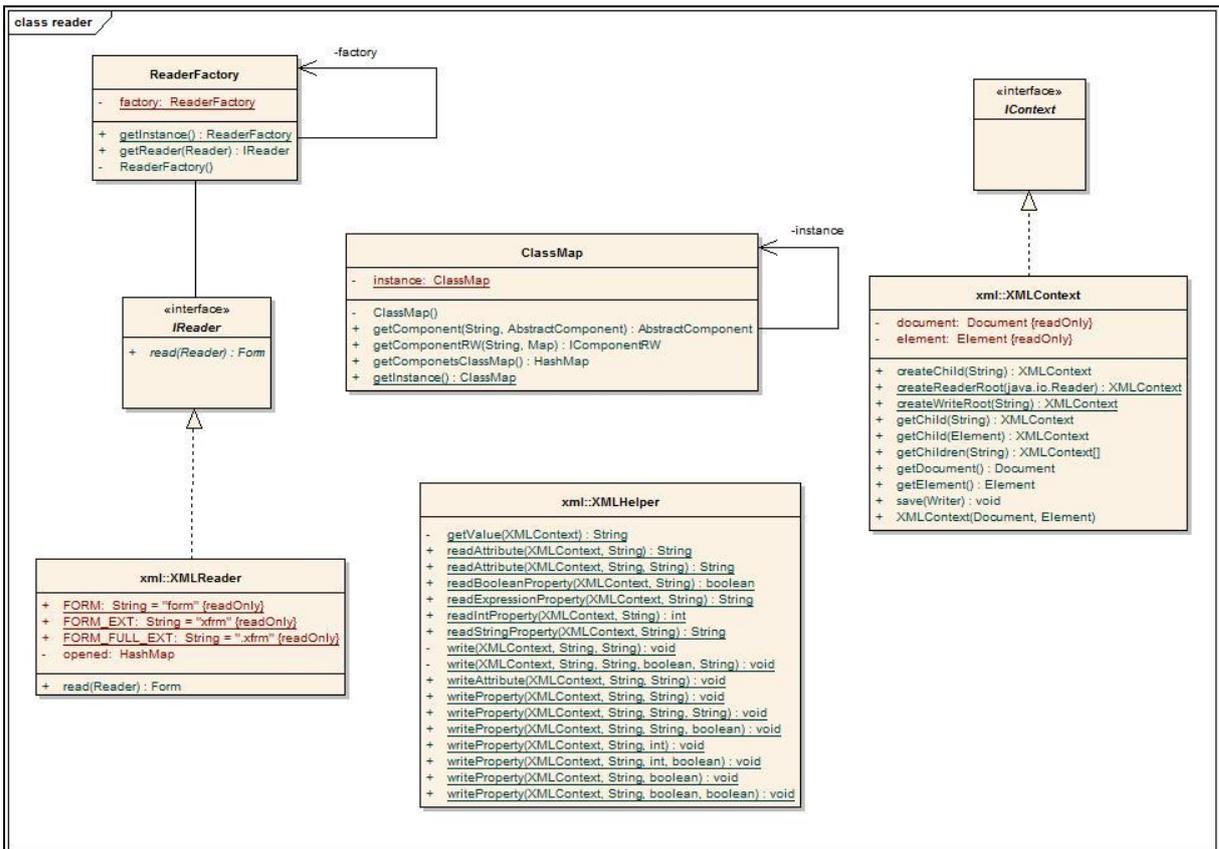


Figura 19 – Diagrama de classe do pacote reader

O sub-pacote `components` possui uma classe correspondente para cada classe do pacote `component`, estas classes são responsáveis pela criação das classes derivadas `AbstractComponent` e por carregar suas variáveis com os valores lidos do arquivo que possui a definição da estrutura da tela.

Algumas classes que deste pacote são `FormRW`, `ButtonRW`, `EditRW` e `LabelRW`, sendo que as classes deste pacote possuem a mesma estrutura hierárquica das classes do pacote `components` (Figura 20).

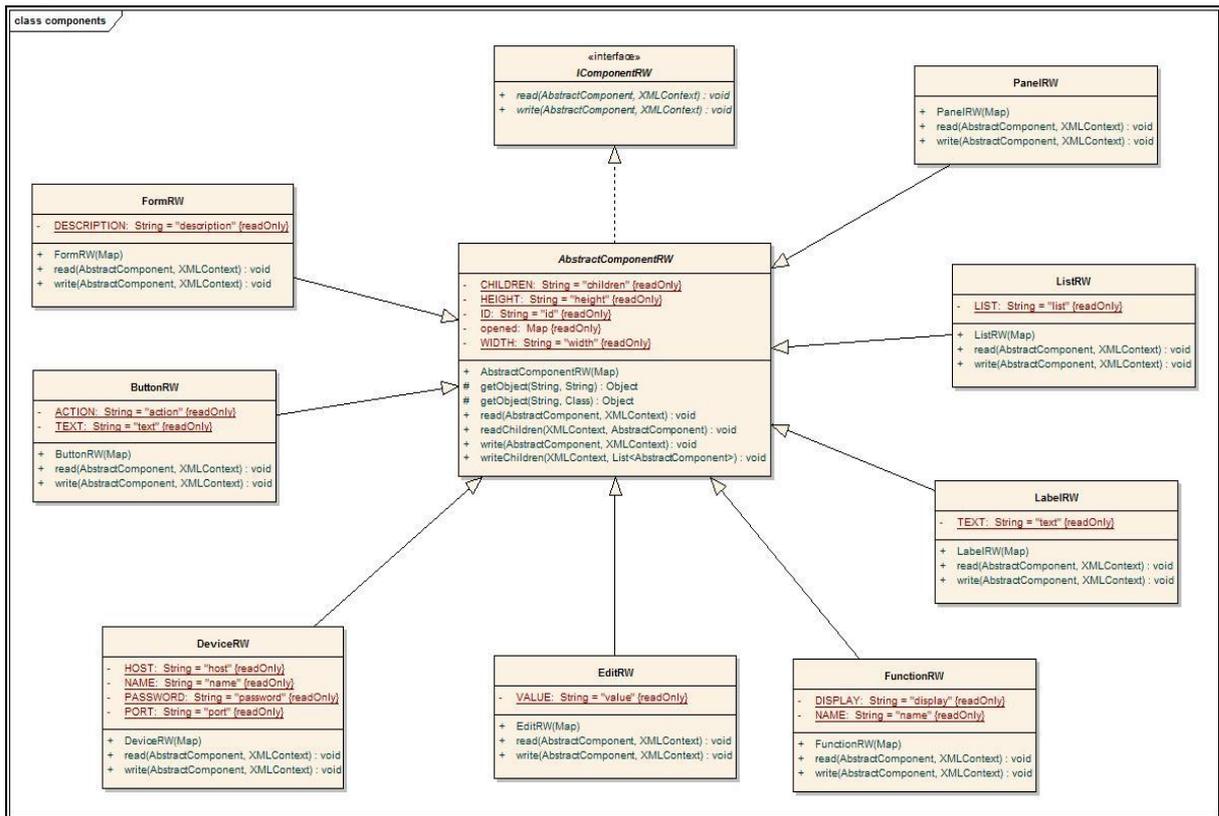


Figura 20 – Diagrama de classes do sub-pacote components

3.2.3 Diagrama de sequência

Esta seção apresenta o diagrama de sequência que representa o conjunto de passos que o programa executa para realizar determinada tarefa, com base nas ações do usuário.

A figura 21 demonstra a sequência de ações realizada para que o usuário realize o caso de uso Cadastrar equipamento.

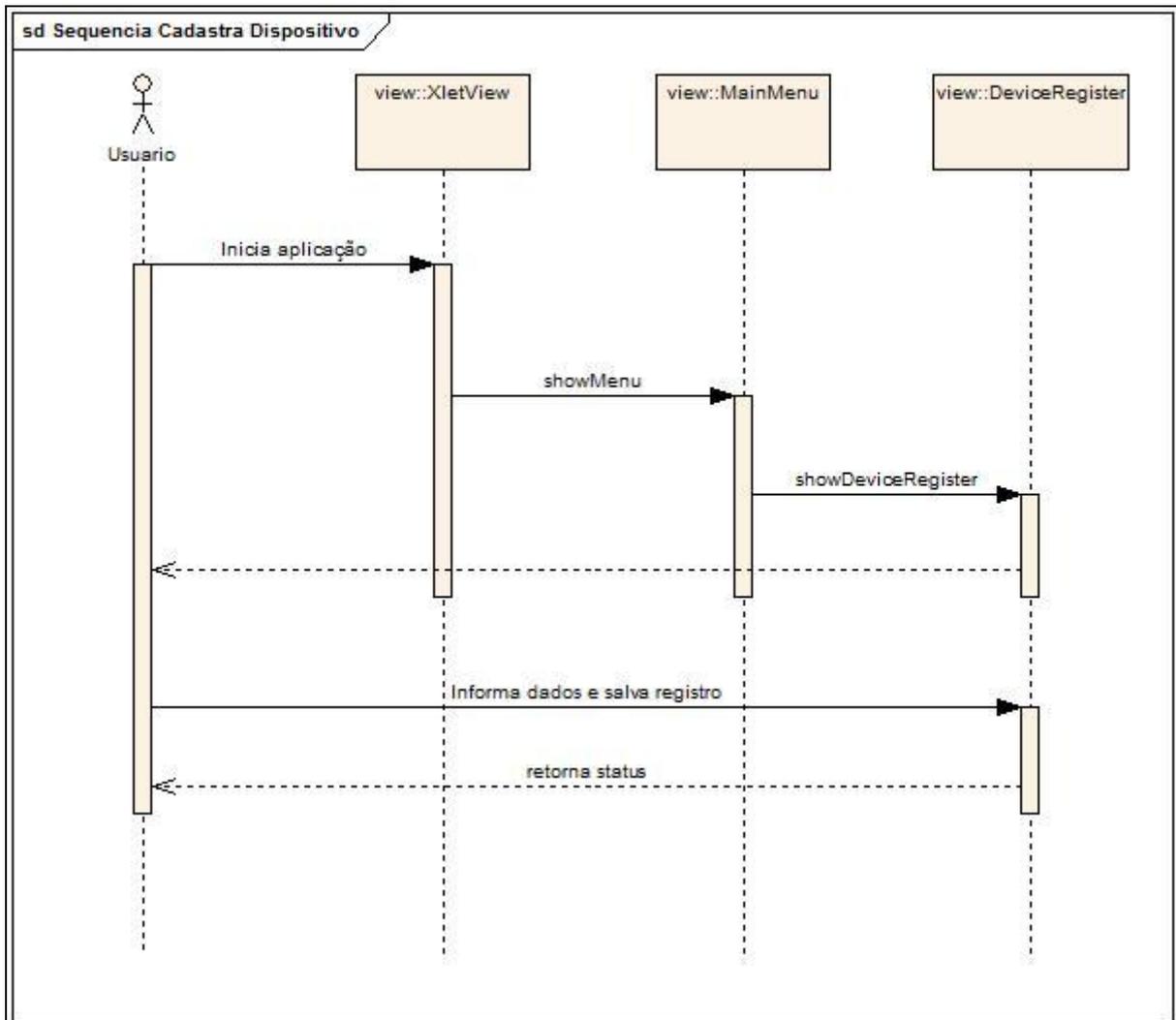


Figura 21 - Diagrama de seqüência do caso de uso Cadastrar equipamento

A figura 22 demonstra a seqüência de ações realizada para que o usuário realize o caso de uso Executar função.

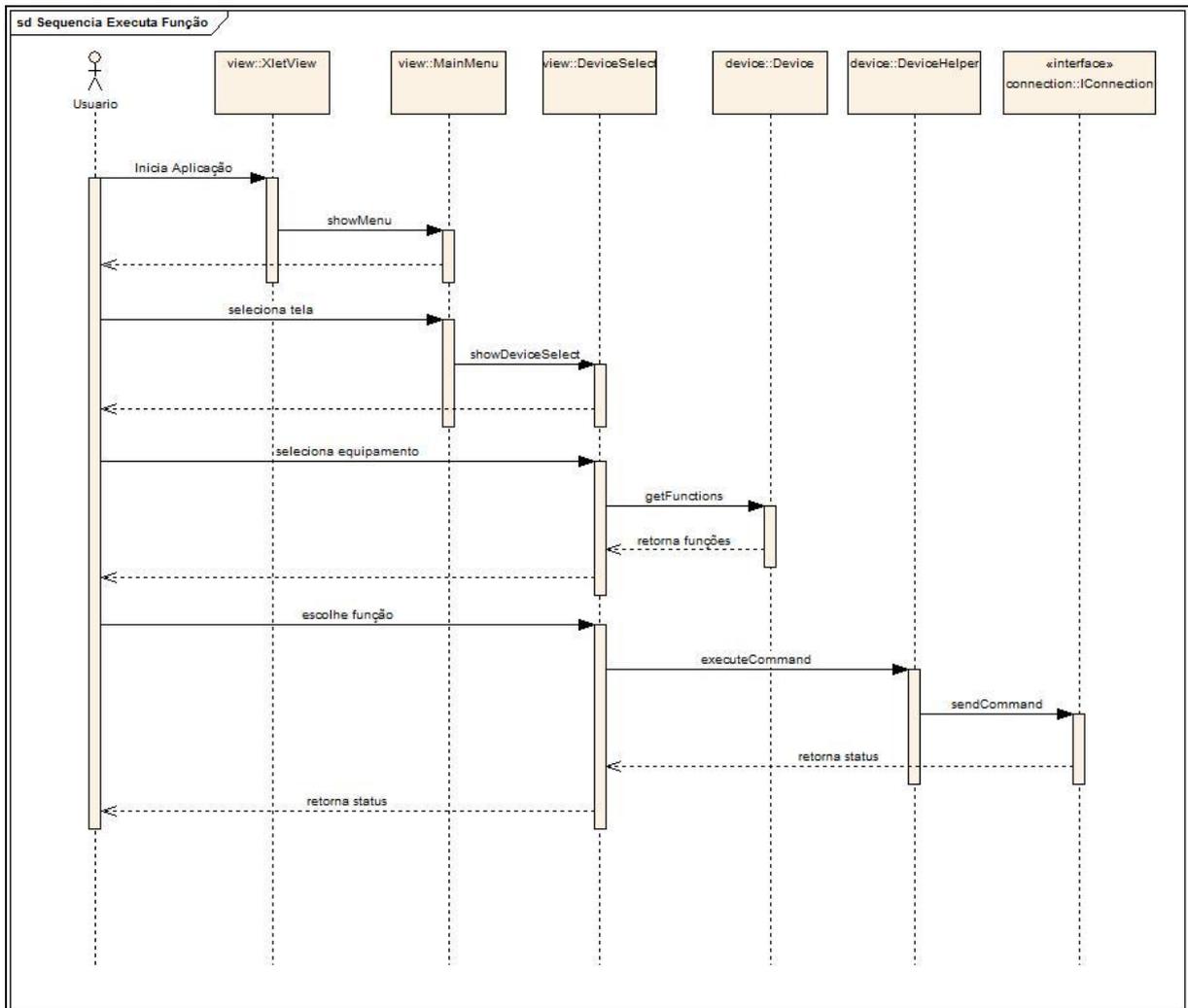


Figura 22 – Diagrama de sequência do caso de uso Executar função

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas de forma detalhada as técnicas e ferramentas utilizadas na implementação do protótipo, bem como questões relacionadas à sua implementação e operacionalidade.

3.3.1 Técnicas e ferramentas utilizadas

O protótipo foi implementado na linguagem Java utilizando o ambiente de desenvolvimento Eclipse. Neste item, são descritos os itens mais relevantes sobre o

desenvolvimento do protótipo.

3.3.1.1 Ambiente de desenvolvimento Eclipse

Como ferramenta de desenvolvimento foi escolhida a *Integrated Development Environment* (IDE) (ECLIPSE 2009) na sua versão 3.5.0. Os motivos para esta escolha são apresentados nesta sessão.

O Eclipse é uma comunidade *open source* cujos projetos estão concentrados na construção de uma plataforma de desenvolvimento aberta e com a principal característica de ser extensível.

Segundo Shavor (2003, p. 4), o Eclipse não é apenas um ambiente de desenvolvimento Java, ele oferece um grande grau de produtividade, flexibilidade e eficiência. A filosofia por traz desta ferramenta é fazer as atividades mundanas consumirem menos tempo provendo ferramentas avançadas que ajudam a gerar, editar e navegar pelo código Java.

O Eclipse ainda não possui nenhum tipo de *plugin* ou adaptação especial para trabalhar com a API JavaTV, porem já possui um *plugin* especial voltado para a criação de programas em Ginga-NCL para a TV Digital Interativa chamado NCL Eclipse.

3.3.1.2 Implementação dos testes

Para implementar os testes do protótipo foi utilizado o *framework* de testes JUnit.

Conforme JUnit (2009), JUnit é um *framework open-source* de desenvolvimento de testes unitários para a linguagem de programação Java.

A versão utilizada foi a 4.7, esta versão utiliza recursos disponíveis a partir da versão 1.5.0 do Java como as *annotations*. O *framework* JUnit é distribuído em um arquivo *jar* (junit-4.7.jar), sendo apenas necessário adicionar o *junit jar* no *classpath* do projeto (MASSOL, HUSTED, 2005).

3.3.1.3 Emulador XletView

O XletView foi o emulador usado para executar Xlets em um PC. Este possui código

aberto sob a licença *General Public License* (GPL) (XLETVIEW, 2003), e a versão utilizada foi a xletview-0.3.6.

Este emulador possui uma implementação de referência da API JavaTV, além de trazer consigo implementações de outras APIs especificadas no padrão MHP, como *Home Audio Video Interoperability* (HAVI) e também a *Digital Audio Video Council* (DAVIC) e implementações especificadas pela própria DVB, além das bibliotecas do *PersonalJava* que o mesmo padrão faz uso (MONTEZ, PICCIONI, 2004).

Ainda segundo Montez e Piccioni (2004), o XletView é desenvolvido em Java sendo sua execução independente do sistema operacional. Este emulador utiliza o Java Media Framework (JMF) 2.1.1, porém com várias deficiências, como a incapacidade de exibir vídeo *Moving Picture Experts Group* (MPEG) relacionado ou controlado por uma Xlet.

Este emulador possui quatro arquivos de configuração (todos ficam dentro do diretório config):

- a) `applications`: é um arquivo XML onde ficam as informações como nome e caminho das Xlets adicionadas no emulador. Caso este arquivo seja modificado é necessário fechar e abrir novamente o emulador para que as alterações tenham efeito. Porém é possível editar esta lista através de uma interface gráfica disponível no menu do emulador em "applications -> Manage applications", desta forma não é necessário recarregar o emulador para que as alterações tenham efeito;
- b) `remote_control`: é um arquivo XML onde ficam as configurações do controle remoto disponível no emulador: é possível alterar propriedades como imagem, posicionamento e *keycode* de cada botão assim como adicionar novos botões;
- c) `channels`: é um arquivo XML onde ficam as configurações de mídia: neste arquivo é possível configurar um vídeo para ser executado como plano de fundo no emulador;
- d) `settings`: é um arquivo texto onde ficam as configurações gerais do emulador. Dentre algumas destas configurações é possível esconder o controle remoto para que a área de vídeo do emulador tenha mais espaço, também é possível adicionar um *classpath* extra ao XletView, desta forma um software Java adicional pode ser acrescentado ao emulador.

A figura 23 apresenta a interface do XletView.



Figura 23 – Tela inicial do XletView

O XletView pode ser executado diretamente a partir do arquivo “xletview.jar”, porém para que seja possível visualizar os resultados de comandos como `System.out.println()` e mensagens de erro, é necessário rodá-lo via console (Figura 24) já que a interface do XletView não possui nenhuma console.

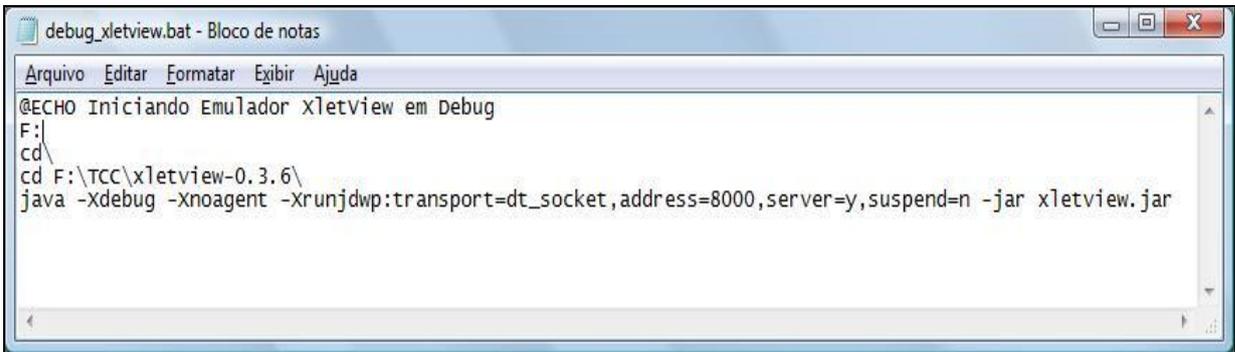
```

C:\Windows\system32\cmd.exe
Iniciando Emulador XletView
C:\Users\rodrigo\Desktop>F:
F:\>cd\
F:\>cd F:\TCC\xletview-0.3.6\
F:\TCC\xletview-0.3.6>java -jar xletview.jar
*****
XletView, Copyright (C) 2003 - 2004 Martin Sveden
XletView comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute
it under certain conditions;
see license document for details.
*****
setting properties...
free/used/total: 2386 K / 5053 K / 7440 K
running gc...
after gc...
free/used/total: 3744 K / 4651 K / 8396 K
[XletView]-INFO->loading Xlet... [Estudo_Xlet]
[XletView]-INFO->XLET started... [Estudo_Xlet]

```

Figura 24 – Console de execução do XletView

Caso seja necessário é possível fazer depuração remota de uma Xlet rodando no emulador, para isto é necessário iniciar o XletView com o comando observado na Figura 25.



```

debug_xletview.bat - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
@ECHO Iniciando Emulador Xletview em Debug
F:\
cd \
cd F:\TCC\xletview-0.3.6\
java -Xdebug -xnoagent -xrunjdpw:transport=dt_socket,address=8000,server=y,suspend=n -jar xletview.jar
  
```

Figura 25 – Comando para ativar o *debug* remoto no XletView

3.3.1.4 Formulário XML

Como a aplicação pode se conectar com os mais diversos tipos de dispositivos é necessário que exista uma abstração na forma como os formulários são transmitidos. A forma escolhida foi o formato XML por ser de fácil interpretação e largamente utilizado na comunicação entre aplicações.

A aplicação irá receber um arquivo no formato XML (Figura 26) que irá conter toda a estrutura de componentes e seus atributos.



```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <form id="FormFc2">
3   <children>
4     <Button id="Bt1">
5       <text>Rango na mesa</text>
6       <action>onclick</action>
7     </Button>
8     <Button id="Bt2">
9       <text>Telefone</text>
10      <action>onclick</action>
11    </Button>
12    <Button id="Bt3">
13      <text>Trovoada</text>
14      <action>onclick</action>
15    </Button>
16  </children>
17 </form>
18
  
```

Figura 26 – Exemplo de arquivo XML interpretado pela aplicação

A partir deste arquivo será gerado um componente Java para cada componente existente no XML. Desta forma, uma árvore de componentes será criada e a partir dela a tela é desenhada de forma dinâmica.

No Quadro 7 pode-se ver um exemplo do código que desenha o componente botão de forma dinâmica a partir do componente gerado do arquivo XML.

```

public Button render(final components.Button com_Button, Container parent) {
    Button button = new Button();
    button.setLabel(com_Button.getText());
    button.setFont(new Font("Arial", Font.BOLD, 20));
    button.setForeground(Color.BLACK);
    button.setBackground(Color.GREEN);
    button.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {
            DeviceHelper.getInstance().executeCommand(com_Button.getAction(),
com_Button, device);
        }
    });
    parent.add(button);
    return button;
}

```

Quadro 7 – Exemplo de rotina que desenha um componente

Para a interpretação do arquivo XML na aplicação é utilizada a API *Document Object Model* (DOM). No Quadro 8 tem-se parte do código da classe `XMLHelper`, uma das classes responsáveis pela leitura do arquivo XML .

```

public class XMLHelper {

    ...

    // Retorna o valor da propriedade
    private static String getValue(XMLContext context) {
        if (context == null) {
            return null;
        }
        Element element = context.getElement();
        NodeList nodes = element.getChildNodes();
        for (int i = 0; i < nodes.getLength(); i++) {
            Node node = nodes.item(i);
            if (node.getNodeType() == Node.TEXT_NODE) {
                return node.getNodeValue();
            }
        }
        return null;
    }
}

```

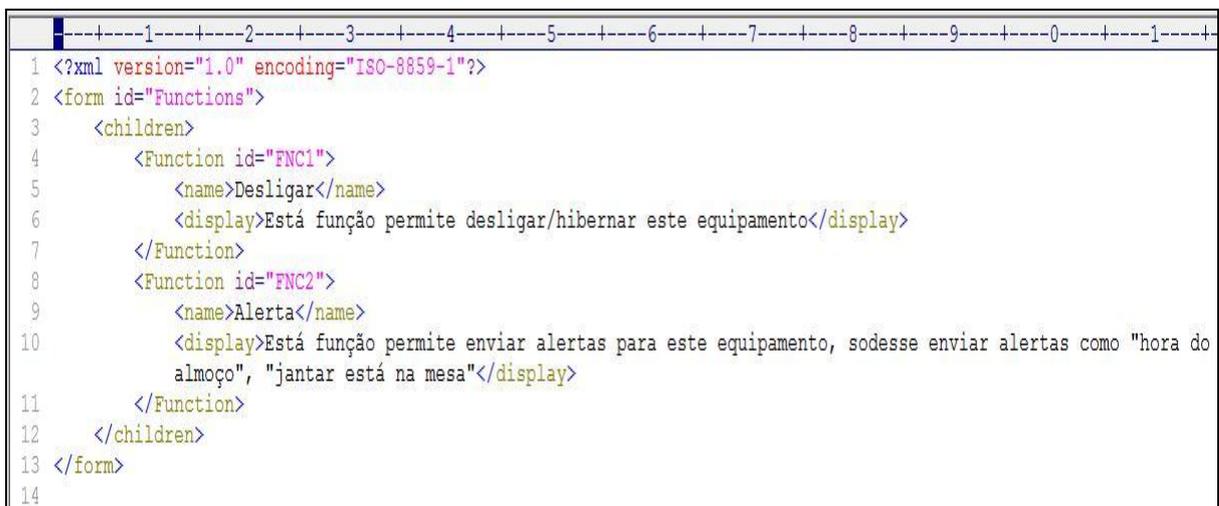
Quadro 8 – Parte do código responsável pela leitura do arquivo XML

3.3.1.5 Protocolo de comunicação

Para que uma aplicação se comunique com os dispositivos conectados a ela, é necessário que um protocolo de comunicação seja definido. Nesta aplicação três comandos são definidos para que um dispositivo se conecte e se comunique, são eles:

- a) `getFunctions`: solicita a lista de funções disponibilizadas por um equipamento. Este comando tem como retorno um arquivo XML (Figura 27) com a definição de cada função disponibilizada;
- b) `getForm:FUNCTION_ID`: solicita o formulário correspondente à função cujo identificador seja igual ao identificador passado no parâmetro `FUNCTION_ID`. Este comando tem como retorno um arquivo XML (Figura 26) com a definição do formulário;
- c) `execFuntion:FORM_ID:ACTION`: executa uma função. Este comando possui dois parâmetros: `FORM_ID` corresponde ao identificador do formulário e `ACTION` a ação executada, por exemplo o identificador do botão clicado. Possui dois possíveis valores de retorno: `OK` se o comando for executado de forma correta e `FAIL` caso contrário.

Todos os comandos e suas respostas são transmitidos entre os delimitadores `BOF` e `EOF`.



```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <form id="Functions">
3   <children>
4     <Function id="FNC1">
5       <name>Desligar</name>
6       <display>Está função permite desligar/hibernar este equipamento</display>
7     </Function>
8     <Function id="FNC2">
9       <name>Alerta</name>
10      <display>Está função permite enviar alertas para este equipamento, sodesse enviar alertas como "hora do
almoço", "jantar está na mesa"</display>
11    </Function>
12  </children>
13 </form>
14

```

Figura 27 – Exemplo de resposta do comando `getFunctions`

3.3.2 Operacionalidade da implementação

Neste item é apresentado o funcionamento do protótipo de software.

Ao executar o protótipo a primeira tela que o usuário irá encontrar será a tela apresentada na Figura 28, onde existe uma mensagem informando o usuário para que ele aperte o botão vermelho do controle remoto para acessar o menu principal da aplicação.

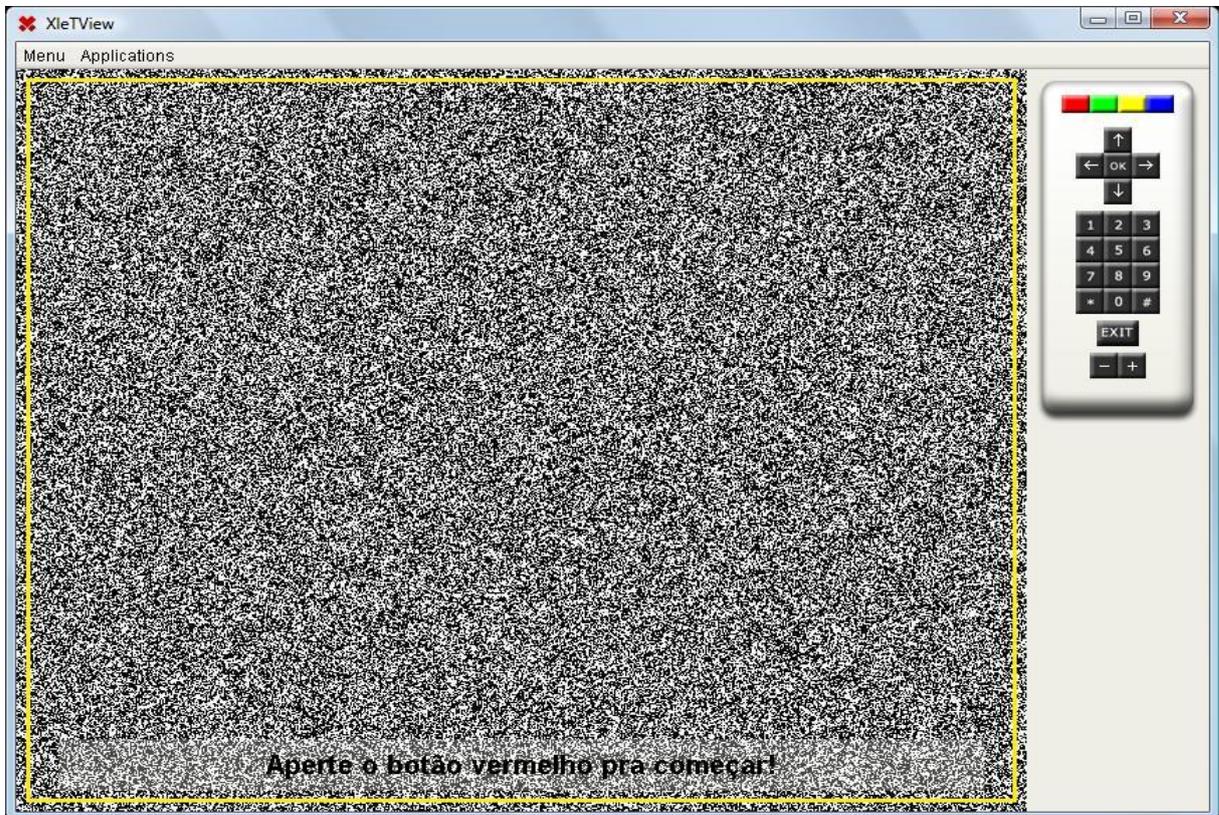


Figura 28 – Tela inicial do protótipo

Ao clicar no botão vermelho do controle a tela do menu (Figura 29) será apresentada para o usuário. Esta tela possui quatro botões, o botão incluir dispositivos, o botão excluir dispositivo, o botão selecionar dispositivo e o botão sair. Caso o usuário queira encerrar a aplicação, além de o botão sair pode apertar o botão azul do controle remoto.

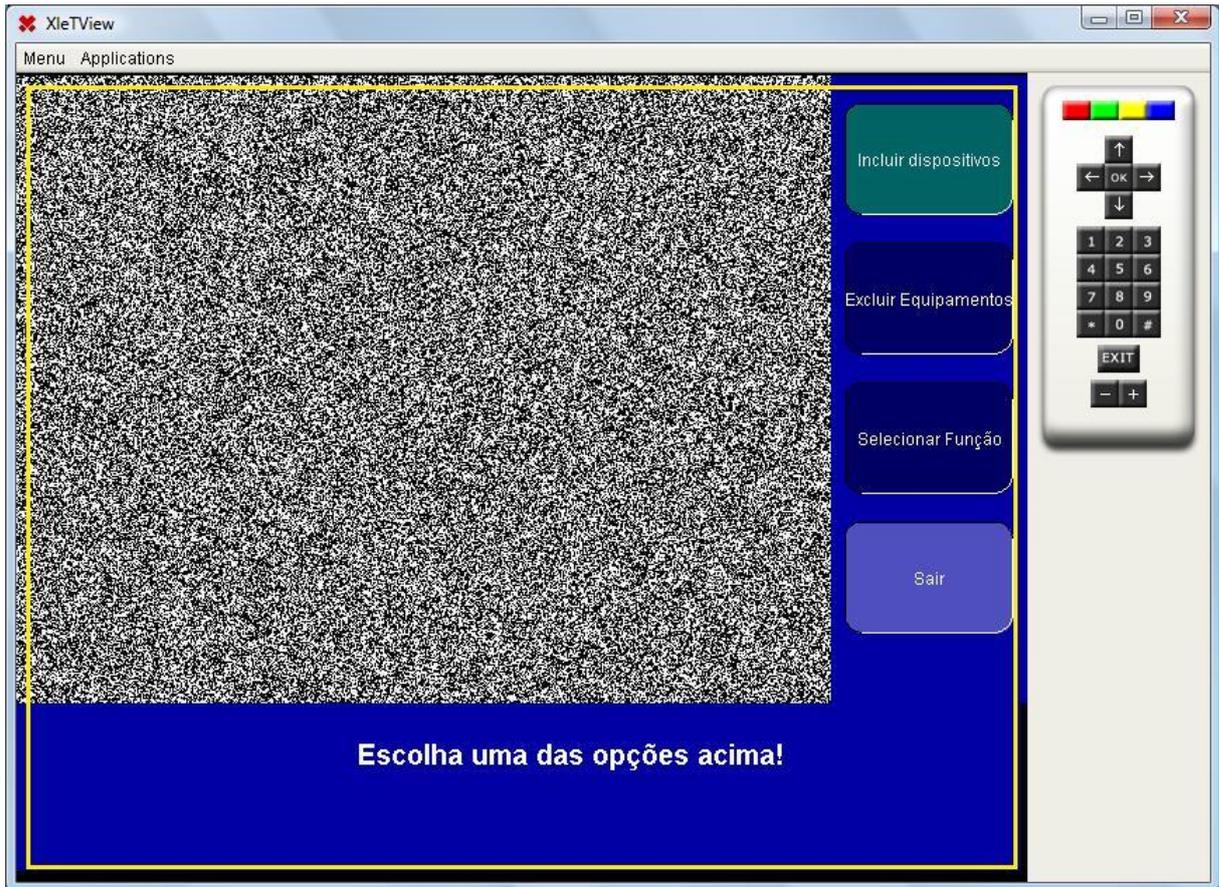


Figura 29 – Menu principal do protótipo

Ao selecionar a opção incluir dispositivo, três campos obrigatórios são exibidos (Figura 30), nome, host e porta. Depois de preencher todos os campos é necessário clicar no botão Gravar para salvar o novo dispositivo. Para retornar ao menu basta clicar no botão Voltar ou no botão azul do controle remoto.

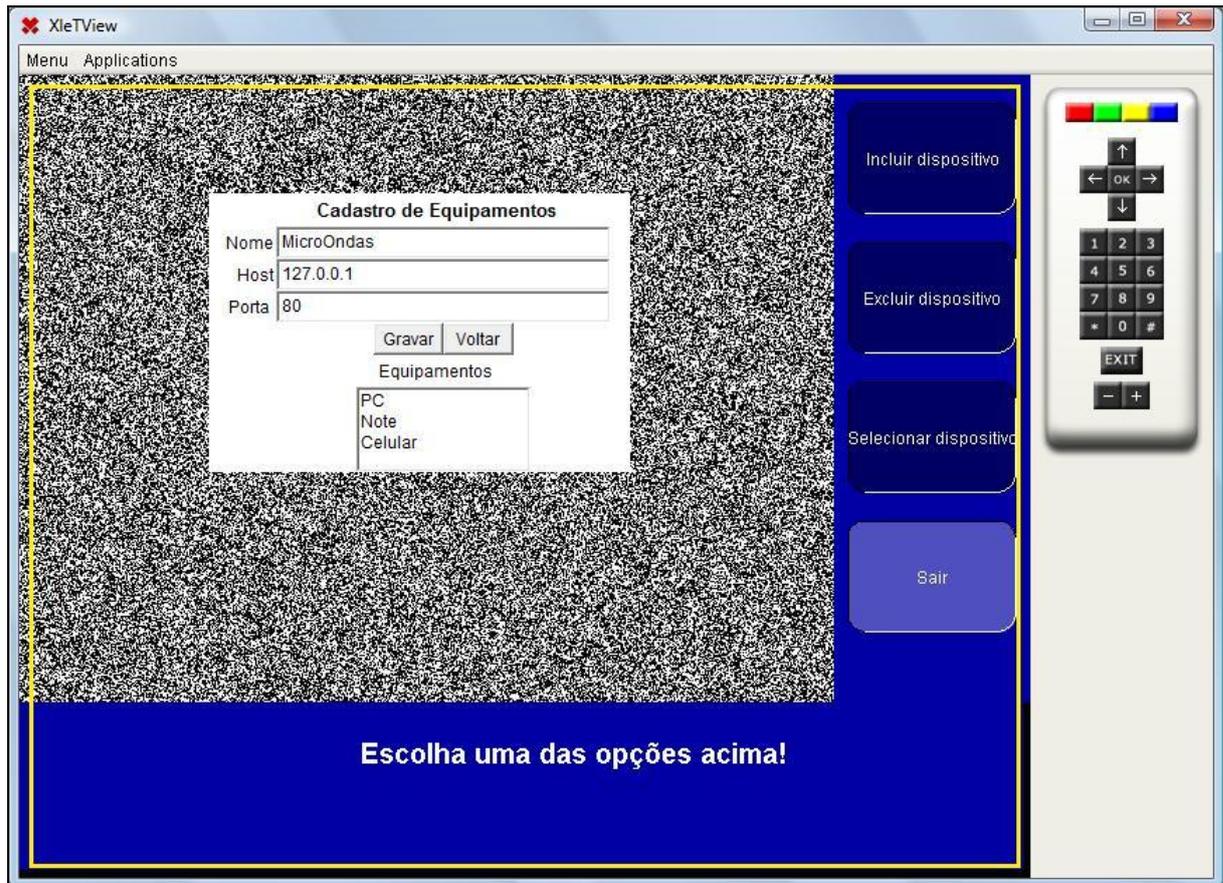


Figura 30 – Tela de inclusão de dispositivos

Para excluir um dispositivo, basta selecionar um equipamento na tela de exclusão (Figura 31) e clicar no botão `Excluir` ou no botão `vermelho` do controle remoto.

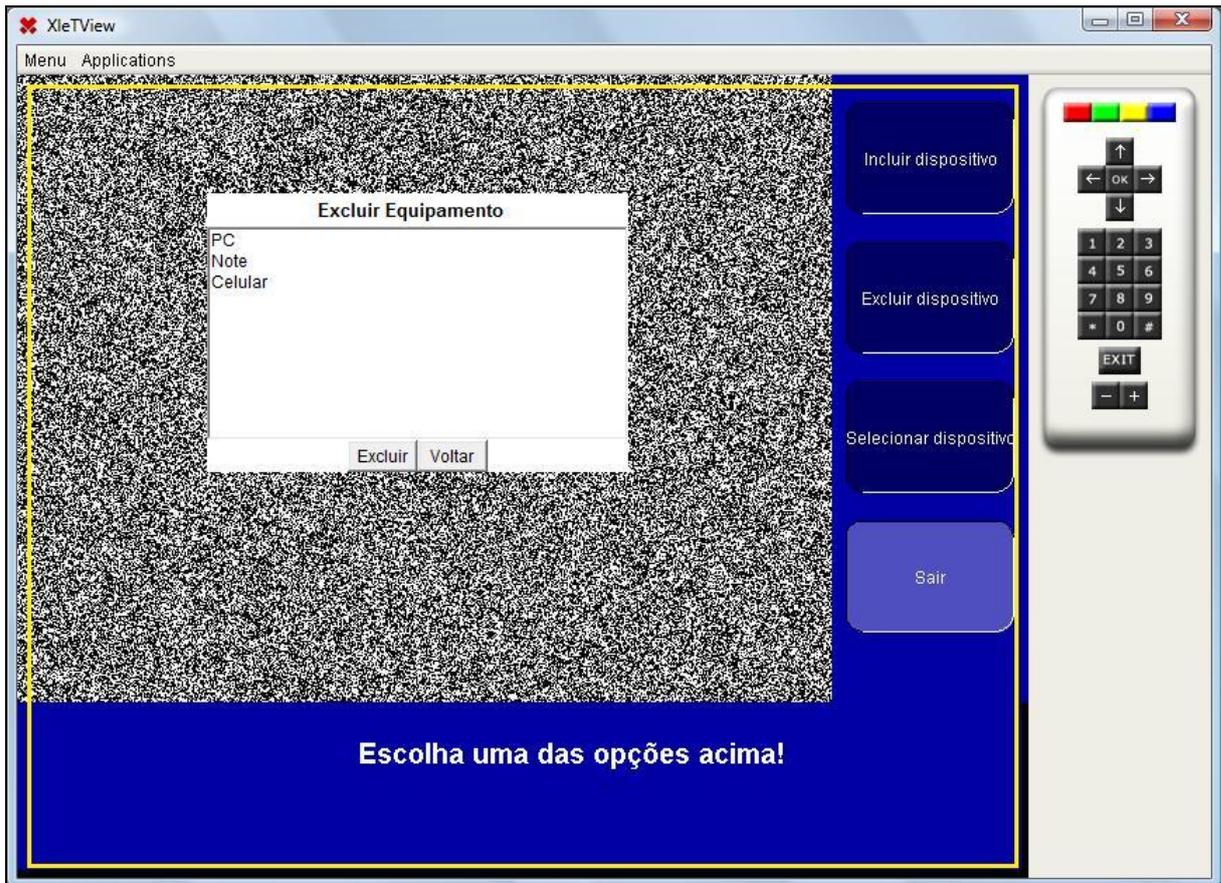


Figura 31 – Tela que exclui um dispositivo

Para executar uma função de um dispositivo cadastrado, o usuário deve acessar o menu *Selecionar dispositivo*, e nesta tela (Figura 32) uma lista com todos os equipamentos cadastrados é exibida para o usuário. Para ativar a seleção é necessário clicar no botão verde do controle remoto ou o botão *Selecionar* da tela.

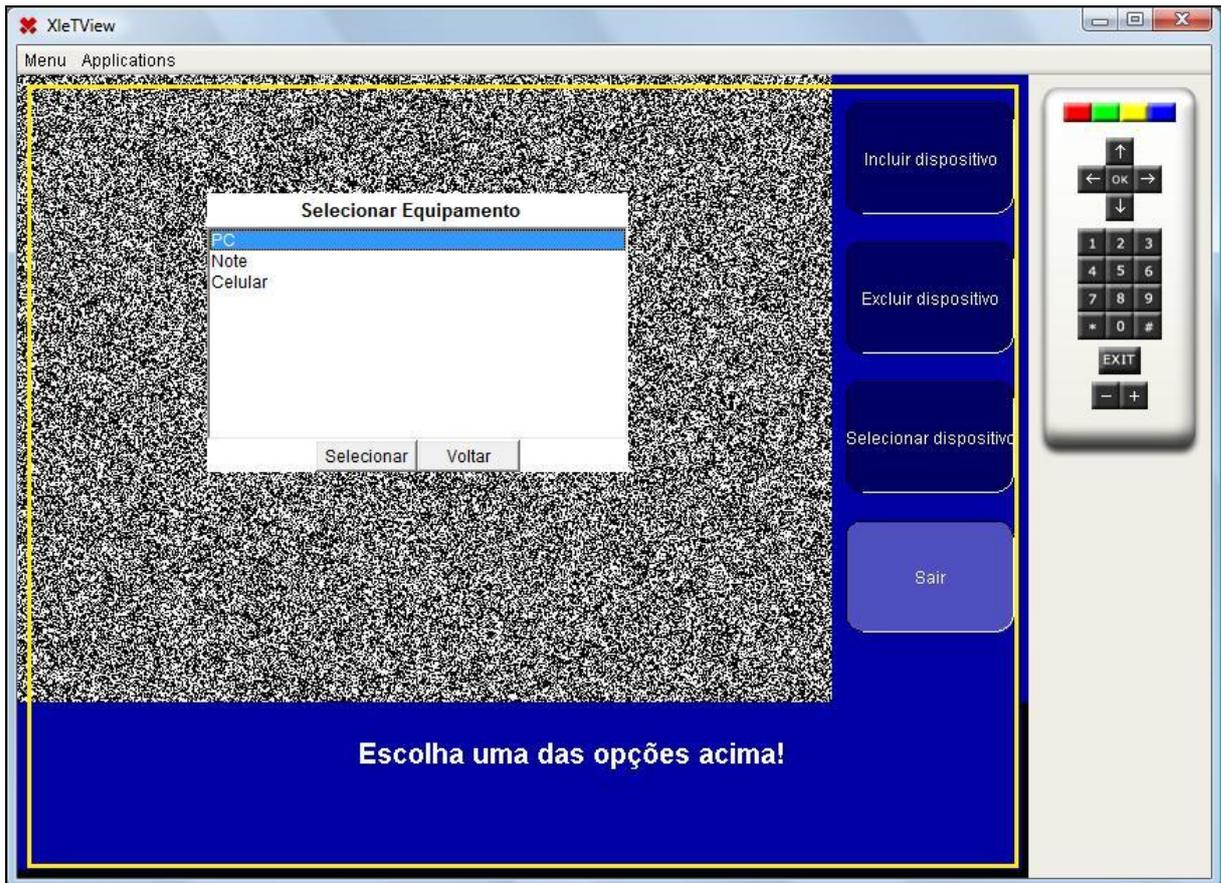


Figura 32 – Tela de seleção de dispositivos

Após selecionar um equipamento, a aplicação requisita ao dispositivo todas as funções que este disponibiliza e exibe ao usuário em uma tela (Figura 33).

No momento em que o usuário estiver visualizando a lista de funções e quiser saber mais detalhes sobre uma função, ele pode acessar esta informação apertando o botão amarelo do controle remoto.

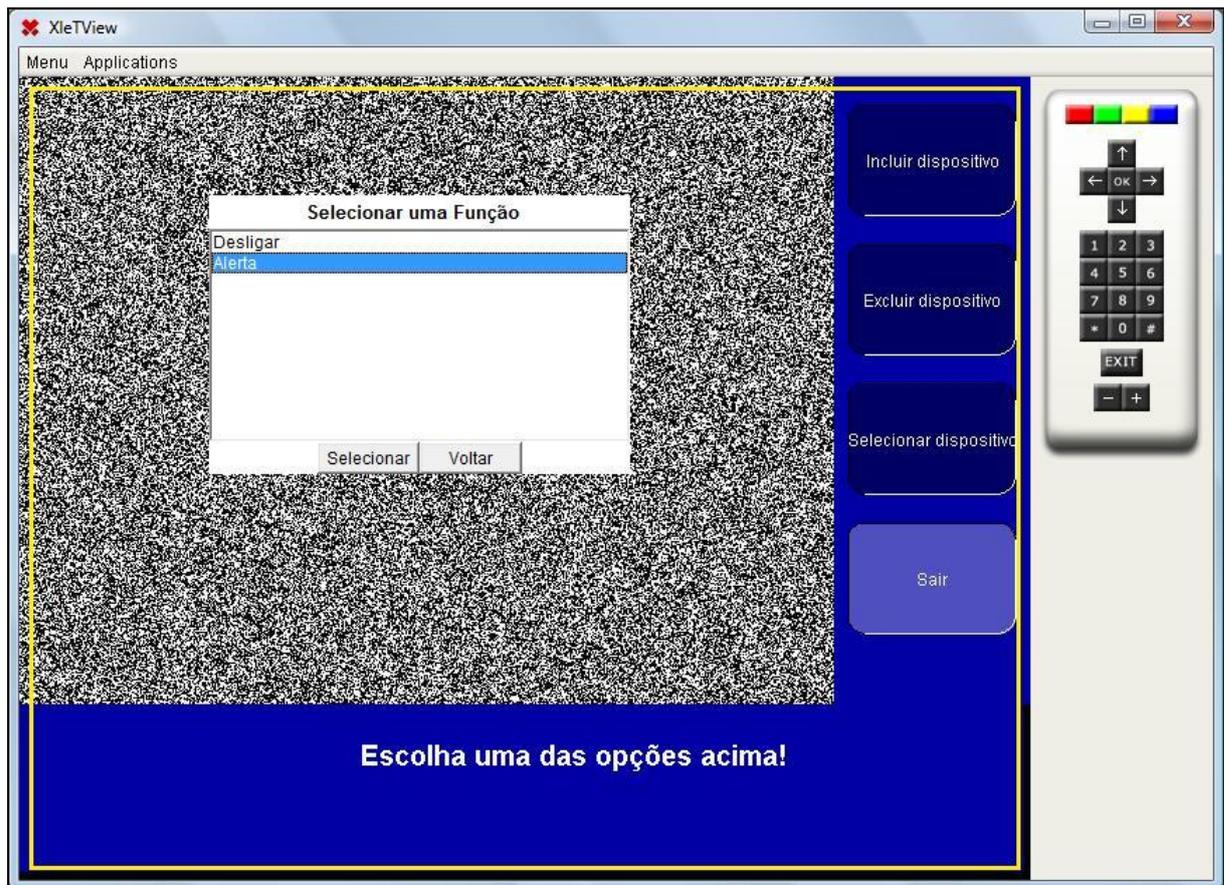


Figura 33 – Tela de seleção de função

Após selecionar uma função, a aplicação requisita ao dispositivo o formulário com as ações desta função, e exibe ao usuário. Este formulário (Figura 34) é dinâmico, pois cada função selecionada possui um formulário diferente.

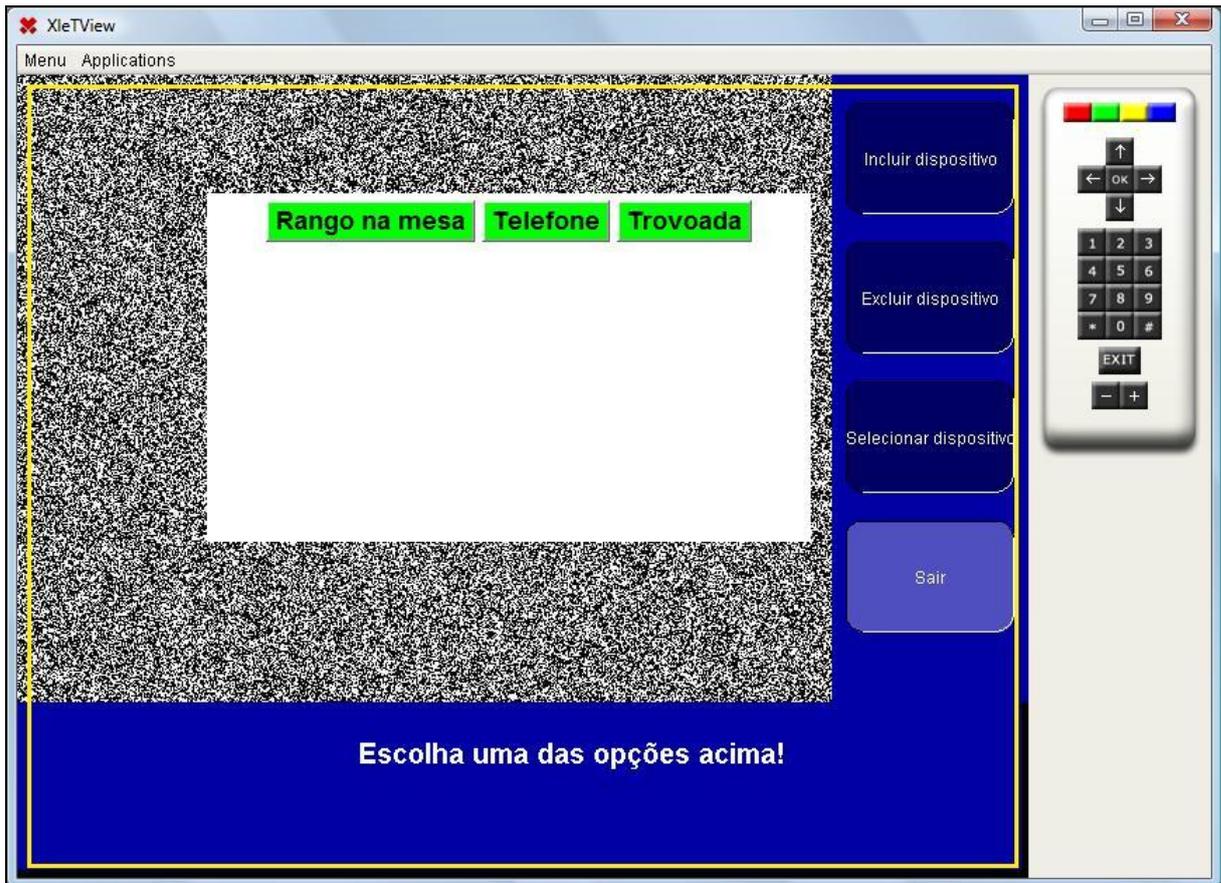


Figura 34 – exemplo de tela com as ações de uma função

3.4 RESULTADOS E DISCUSSÃO

Os resultados encontrados com o término do trabalho são satisfatórios, pois com a utilização do protótipo foi possível provar o conceito de inter-comunicação dos dispositivos presentes em uma residência através da tecnologia da TV digital, portanto alcançando o objetivo inicial.

Todos os três objetivos específicos foram atingidos.

Para identificar um equipamento ligado a uma rede local TCP/IP foi criado um cadastro de equipamentos, onde é informado o IP do equipamento e a porta disponibilizada por este para comunicação com o protótipo.

Foi proposto um protocolo de comunicação e adotado o formato XML para envio dos dados, desta forma foi atingida abstração necessária para comunicação entre os mais diversos tipos de equipamentos, abstraindo inclusive sistemas operacionais e linguagens de

programação.

O último objetivo foi atingido em consequência da realização dos objetivos anteriores, com a conexão e um protocolo de comunicação entre os equipamentos estabelecida, o controle de qualquer função pode ser implementado de acordo com a criatividade do desenvolvedor.

Santos Neto (2005) direcionou os esforços de sua pesquisa sobre tecnologias de comunicação inter-aplicações para uma solução que permitia comunicação entre dispositivos semelhantes. Apenas alguns testes de conceito foram realizados com a finalidade de atestar o funcionamento adequado da combinação de *web services* e *bluetooth* em um ambiente embarcado.

A aplicação comercial de controle de habitações inteligentes desenvolvida por Mordomus (2008) integra todos os dispositivos conectados ao sistema, permitindo a gestão e o controle de várias funções e configurações da casa a partir de uma central de controle ou acesso remoto a partir de uma PDA. Entre os trabalhos correlatos apresentou-se como ótima base conceitual.

4 CONCLUSÕES

De forma geral, o trabalho atingiu o objetivo de desenvolver e demonstrar o funcionamento de uma aplicação interativa para TV digital que pudesse controlar algumas funções de equipamentos presentes em uma residência conectados a uma rede TCP/IP. Todos os objetivos específicos foram atingidos com sucesso.

A escolha do ambiente de desenvolvimento foi feita levando em consideração as dificuldades mencionadas, tendo sido optado por utilizar o ambiente XletView. Mesmo existindo algumas limitações comuns em emuladores, o XletView mostrou-se satisfatório em relação ao que a ferramenta propõe-se a fazer: simular o funcionamento de uma TV digital através dos recursos da API Java TV e do padrão MHP.

Fica evidente o grande potencial da interatividade da TV digital. Este “novo computador” que se terá em casa pode ser amplamente explorado, inclusive utilizando este ambiente para desenvolver aplicações de domótica.

Algumas dificuldades foram encontradas durante o desenvolvimento do trabalho, entre elas a principal foi o atraso na entrega no *middleware* Ginga-J, já que inicialmente a idéia era desenvolver a aplicação utilizando este *middleware*. Este atraso contínuo, devido a disputas relacionadas a patentes de parte do código Java utilizado, vem frustrando toda uma comunidade acadêmica que anseia pela chegada deste padrão tão amplamente divulgado para começar a realizar pesquisas e estudos nessa área.

Outra dificuldade encontrada foi a falta de material didático que abordasse o assunto. Ainda existem poucos livros que abordam a realidade brasileira, boa parte do material utilizado teve como fonte a internet, assim como há ainda uma falta de aplicações comerciais que pudessem servir como fonte de inspiração. Para resolver este problema, foi necessária uma visita à fundação CERTI em Florianópolis, onde é desenvolvido tanto o software como o hardware para TV digital. Com esta visita várias dúvidas principalmente referentes ao contexto do SBTVD foram sanadas.

Este protótipo possui como limitação a pouca quantidade de componentes e o *layout* pouco desenvolvido, porém toda a aplicação foi desenvolvida tendo em mente a adição de novas funcionalidades, tendo seu núcleo bem estruturado e independente, bastando apenas adicionar novas funcionalidades ao protótipo.

4.1 EXTENSÕES

Existem vários pontos que podem ser melhorados na aplicação, os quais são:

- a) testar e converter a aplicação para GINGA-J assim que este estiver oficialmente definido e liberado;
- b) desenvolver um editor gráfico para a construção dos formulários, permitindo que qualquer usuário seja capaz de produzir conteúdo para a aplicação;
- c) aprimorar questões relacionadas à usabilidade e à interface com o usuário;
- d) implementar suporte a outras formas de conexão como *web services* e *bluetooth*;
- e) analisar a parte de segurança das transmissões entre a aplicação e os demais dispositivos.

REFERÊNCIAS BIBLIOGRÁFICAS

ANGEL, Patricia M.; FRAIGI, Liliana B. **Introducción a la domotica, tomo I**. Embalse: EBAI, 1993. 172 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Projeto 00:001.85-006/4**: informação e documentação: referências – elaboração. Rio de Janeiro, 2007. Norma ainda não homologada.

BRUGNERA, Mauro R. **Domótica**. [S.l.], 2006. Disponível em: <<http://www.unibrattec.com.br/jornadacientifica/diretorio/FEEVALE+MRB.pdf>>. Acesso em: 21 mar. 2009.

CETIC.BR. **Pesquisa sobre o uso das tecnologias da informação e da comunicação no brasil**. [S.l.], 2007. Disponível em: <<http://www.cetic.br/tic/2007/indicadores-cgibr-2007.pdf>>. Acesso em: 21 mar. 2009.

COELHO, Adriano S. S. **Uma especificação de desenvolvimento de serviços para televisão digital interativa**. 2005. 112 f. Trabalho de Conclusão de Curso - (Graduação em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2005.

CROCOMO, Fernando. **TV digital e produção interativa: a comunidade manda notícias**. Florianópolis: Ed. da UFSC, 2007. 178 p.

DTV. [S.l.], 2009. Disponível em: <<http://www.dtv.org.br/materias.asp?id=233&menuid=3>>. Acesso em: 20 out. 2009.

Eclipse. [S.l.], 2009. Disponível em: <<http://www.eclipse.org>>. Acesso em: 30 out. 2009.

FREITAS, Igor V. B. **Televisão digital: que imagem terá o modelo brasileiro**. Brasília, 2004. 64 f. Disponível em: <http://www.senado.gov.br/conleg/textos_discussao/texto18%20-%20igor.pdf>. Acesso em: 1 out. 2009.

FUNDAÇÃO CERTI. **HD-ONE**: a melhor e mais versátil plataforma de set-top box para TV digital de alta definição. Florianópolis, 2008. Folheto.

GINGA. [S.l.], 2009. Disponível em: <<http://www.ginga.org.br/>>. Acesso em: 20 mar. 2009.

GUIMARÃES, Rangner F. **Interação do espectador com TV digital**. 2005. 52 f. Trabalho de Graduação (Bacharelado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2004-2/rfg.doc>>. Acesso em: 30 set. 2009.

HDX. [S.l.], 2009. Disponível em:

<http://www.hxd.com.br/site/index.php?option=com_content&task=view&id=18&Itemid=69>. Acesso em: 20 out. 2009.

JUNIT. [S.l.], 2009. Disponível em: <<http://www.junit.org>>. Acesso em: 30 out. 2009.

LEMOS, André L. M. **Anjos interativos e retribuição do mundo: sobre interatividade e interfaces digitais.** [S.l: s.n.], 1997.

MASSOL, Vincent; HUSTED, Ted. **JUnit em ação.** Rio de Janeiro: Ciência Moderna, 2005. 404 p.

MINISTÉRIO DAS COMUNICAÇÕES. **Sistema brasileiro de TV digital.** Brasília, 2003. Disponível em: <<http://sbtvd.cpqd.com.br>>. Acesso em: 30 set. 2009.

MONTEZ, Carlos; BECKER, Valdecir. **TV digital interativa: conceitos, desafios e perspectivas para o Brasil.** 2. ed. rev. e ampl. Florianópolis: Ed. da UFSC, 2005. 200 p.

MONTEZ, Carlos; PICCIONI, Carlos A. **Um estudo sobre emuladores de aplicações para a televisão digital interativa.** Florianópolis, 2004. Disponível em: <<http://www.tvdi.inf.br/upload/artigos/artigo7.pdf>>. Acesso em: 21 set. 2009.

MORDOMUS. **Intelligent house management.** [S.l.], 2008. Disponível em: <<http://www.mordomus.com>>. Acesso em: 21 mar. 2009.

MORRIS, Steven; SMITH-CHAIGNEAU, Anthony. **Interactive TV standards.** Burlington: Focal Press, 2005. 585 p.

PORTAL DO SOFTWARE PÚBLICO BRASILEIRO. [S.l.], 2009a. Disponível em: <<http://www.softwarepublico.gov.br/dotlrn/clubs/ginga>>. Acesso em: 20 mar. 2009.

_____. [S.l.], 2009b. Disponível em:

<<http://www.softwarepublico.gov.br/dotlrn/clubs/ginga/file-storage/view/public/posters/Poster2.pdf>>. Acesso em: 20 mar. 2009.

SANTOS NETO, Jayro J. dos. **Um estudo sobre comunicação inter-aplicações em um ambiente de TV digital.** 2005. 101 f. Trabalho de Conclusão de Curso - (Graduação em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2005.

SHAVOR, Sherry. **The Java developer's guide to Eclipse.** Boston, MA: Addison-Wesley, 2003. 854 p.

SILVA, Jones Q. da. **TV digital interativa.** 2003. 56 f. Monografia (Especialização em Redes de Computadores) – Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2003.

SOARES, Luiz F. G.; BARBOSA, Simone D. J. **TV interativa se faz com Ginga**. Rio de Janeiro, 2006. Disponível em: <<http://www.tvdi.inf.br/upload/artigos/jai2008-ppt.pdf>>. Acesso em: 30 set. 2009.

STEUER, Jonathan. Defining virtual reality: dimensions determining telepresence. **Journal of Communication**, v. 42, n. 4, p. 73-93, fall, 1992.

SUN MICROSYSTEMS. **JavaTV API specification**. [S.l.], 2001. Disponível em: <<http://java.sun.com/products/javaTV>>. Acesso em: 27 set. 2009.

VENTURI, Eli. **Protótipo de um sistema para controle e monitoração residencial através de dispositivos móveis utilizando a plataforma .NET**. 2005. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2005.

XLETVIEW. [S.l.], 2003. Disponível em: <<http://www.xletview.org>>. Acesso em: 30 out. 2009.