

UNIVERSIDADE FEDERAL DO AMAZONAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

IMPLEMENTAÇÃO DE SERVIÇOS DE CUIDADOS COM A
SAÚDE (*HEALTHCARE*), INTEGRADOS EM UM
GATEWAY COMPATÍVEL COM O MODELO BRASILEIRO
DE TV DIGITAL

VANDERMI JOÃO DA SILVA

MANAUS - AM

2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DO AMAZONAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

VANDERMI JOÃO DA SILVA

IMPLEMENTAÇÃO DE SERVIÇOS DE CUIDADOS COM A
SAÚDE (*HEALTHCARE*), INTEGRADOS EM UM
GATEWAY COMPATÍVEL COM O MODELO BRASILEIRO
DE TV DIGITAL

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Informática, área de concentração: Engenharia de Software

Orientador:

Prof. Dr. -Ing. Vicente Ferreira de Lucena Junior

MANAUS - AM

2010

VANDERMI JOÃO DA SILVA

IMPLEMENTAÇÃO DE SERVIÇOS DE CUIDADOS COM A
SAÚDE (*HEALTHCARE*), INTEGRADOS EM UM
GATEWAY COMPATÍVEL COM O MODELO BRASILEIRO
DE TV DIGITAL

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Informática, área de concentração: Engenharia de Software

Aprovado em 2010.

BANCA EXAMINADORA

Prof. Dr. -Ing. Vicente Ferreira de Lucena Junior
Universidade Federal do Amazonas - UFAM

Prof. Dr. José Pinheiro de Queiroz Neto
Instituto Federal de Educação Ciência e Tecnologia do Amazonas - IFAM

Prof. Dr. João Edgar Chaves Filho
Universidade Federal do Amazonas - UFAM

Dedico este trabalho à minha família, em especial à três pessoas muito importantes na minha vida, minha mãe Nair Antônia da Silva, pelo exemplo de perseverança e honestidade, minha esposa Nilda Macedo da Silva, pela compreensão e pelo apoio moral me dado durante as fases mais difíceis enfrentadas no decorrer do trabalho e minha filha Alexia Macedo da Silva pelas vezes que me acompanhou durante os experimentos.

Agradecimentos

À Deus o grande criador que me oferece o ar que eu respiro e minha saúde perfeita para suportar as adversidades da vida.

À meus familiares, pelo apoio me dado durante esse trabalho e por acreditarem no meu potencial.

Aos meus amigos do grupo de TV Digital e do Mestrado em Engenharia Elétrica e Informática, Lady Daiana, Gisele Lira, Jomara Bindá, Charles Melo, Orlewisson Maia, Luciano Pinto, Antonio Santos, Nairon Viana, Ricardo Erikson, Alexandre Martiniano, Alexandre Pereira e Marcelo Oliveira, pelo apoio e pelos momentos de descontração vividos durante essa caminhada.

Ao Diretor do Instituto de Ciências Exatas de Itacotiara (UFAM-ICET), professor Cícero Mota Cavalcante, pelo apoio dado a minha formação profissional.

Aos meus alunos do curso de Sistemas de Informação do ICET-UFAM especialmente às turmas de Banco de Dados, Engenharia de Software e Sistemas de Informação e Sociedade, que me apoiaram quando precisei me ausentar da sala de aula em virtude de reuniões com o meu orientador.

A todos os professores e coordenação técnica do Programa de Mestrado em Informática da UFAM por contribuírem direta e indiretamente para minha formação acadêmica.

Ao professor Raimundo Barreto, coordenador do Programa de Mestrado em Informática pelo apoio dado durante o trabalho.

Ao orientador, professor Vicente Ferreira de Lucena Junior, por todo suporte durante o Mestrado, principalmente pelas cobranças dos artigos científicos, que contribuíram para minha formação acadêmica.

Ao CNPq, CAPES, FAPEAM, UFAM e CETELI, pelo apoio financeiro através de bolsa de estudo e laboratórios pra desenvolvimento da pesquisa.

*Há homens que lutam um dia e são bons.
Há outros que lutam um ano e são
melhores. Há os que lutam muitos anos e
são muito bons. Porém, há os que lutam
toda a vida. Esses são os imprescindíveis.*

Bertolt Brecht

Resumo

A TV digital interativa é uma tecnologia que usa a modulação e compressão de vídeo, áudio e sinais de dados e os envia a aparelhos compatíveis com a tecnologia, proporcionando desta maneira a transmissão e recepção do conteúdo em um mesmo canal, com imagens de alta definição, qualidade e interatividade. O padrão brasileiro de TV digital está sendo implantado no Brasil, com término da implantação previsto para 2016. O conceito de *Healthcare*, trata sobre cuidados com a saúde, prevenção de doenças, qualidade de vida dentre outras, e as aplicações de *healthcare*, disponíveis no mercado, atendem a serviços hospitalares, casas de saúde de idosos dentre outros serviços. A integração das tecnologias de *healthcare* com as redes residenciais estão cada vez mais presentes no cotidiano das pessoas, portanto com o intuito de investigar a integração de dispositivos médicos convencionais e propor uma arquitetura compatível com a TV digital brasileira, este trabalho apresenta duas implementações de protótipos para medição de frequência cardíaca e pressão arterial através de um *gateway* residencial, que intermedia a coleta de dados entre dispositivos e a TV. Como resultado, a arquitetura do *gateway* é definida, bem como a forma de acesso a uma base de dados comum aos dispositivos conectados. Os teste foram feitos usando um emulador e dois *Set Top boxes* sendo um compatível com o padrão brasileiro de TV digital e outro compatível com o padrão europeu.

Palavras-chave: Ginga, *Gateway* Residencial, *Healthcare*, TV Digital Interativa.

Abstract

The digital interactive TV is a technology that uses the modulation and video compression, audio and data signals and sends them to compatible devices with the technology, providing this way the transmission and reception of content in a single channel, with high definition images , quality and interactivity. The digital TV standard is implemented in Brazil and the total deployment is scheduled for 2016. The concept of “healthcare”, is about the health care, disease prevention, quality of life among others. The healthcare applications available on the market, meet hospital services, nursing homes for the elderly among other services. The integration of healthcare technologies with home networks are present in daily life, so in order to investigate the integration of conventional medical devices and propose an architecture compatible with digital TV in Brazil this thesis presents two implementations of prototypes for heart rate and blood pressure measure using a residential gateway for collecting data between devices and the TV. As a result, the gateway architecture is defined with the access a common database to connected devices. The tests were made using an emulator and two set top boxes compatible digital TV standard and one compatible with the European standard.

Keywords: Ginga,Residential Gateway, Healthcare, Digital Interactive Television.

Lista de Siglas

AI	Agentes Inteligentes
AmI	Ambient Intelligence
ANATEL	Agência Nacional de Telecomunicações
API	Application Programming Interface
BSN	Body Sensor Network
CETELI	Centro de Tecnologia Eletrônica e Telecomunicações
MIDP	Connected Limited Device Configuration
CODEC	Codificadores e Decodificadores de Vídeo
GEM	Globally Executable MHP
GINGA	Middleware Brasileiro de TV Digital
GINGA-J	Subsistema Procedural para a TV digital brasileira
GINGA-NCL	Subsistema Declarativo para a TV Digital Brasileira
GPRS	General Packet Radio Service
GR	Gateway Residencial
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
IMC	Índice de Massa Corporal
IP	Internet Protocol
ISDB-T	Integrated Services Digital Broadcasting Terrestrial
ISDB-TB	Integrated Services Digital Broadcasting Terrestrial Brazilian
ISTAG	Information Society Technologies Advisory Group
JME	Java Micro Edition
JVM	Java Virtual Machine
KVM	Kilobyte Virtual Machine
MHP	Multimídia Home Platform

MIDP	Mobile Information Device Profile
MPEG-2	Padrão de Codificação para Vídeo e Áudio Digital
NCL	Nested Context Language
OSGI	Open Services Gateway Initiative
OWL	Web Ontology Language
PDA	Personal Digital Assistant
POO	Programação Orientada a Objetos
RFID	Radio Frequency Identification
RTOS	Real-time Operating System
RXTXJava	Biblioteca escrita em Java para transmissão e recepção de dados via porta serial
SMS	Short Message System
SMSLib	Biblioteca SMS
STB	set-top box
TVDi	Televisão Digital Interativa
USB	Universal Serial Bus
WEB	Rede de computadores de alcance mundial
XBee	Sensor que utiliza a tecnologia Zigbee
XML	Extensible Markup Language

Lista de Figuras

Figura 1.1	Etapas da Metodologia de Desenvolvimento do Trabalho.	21
Figura 2.1	Arquitetura do <i>middleware</i> brasileiro GINGA.	24
Figura 2.2	Arquitetura da API Java TV.	25
Figura 2.3	Ciclo de Vida de uma Aplicação Xlet	26
Figura 2.4	Exemplo de um Arquivo NCL	27
Figura 2.5	Visão Geral de uma Infra-estrutura de Redes Residenciais Usando um GR	27
Figura 3.1	Arquitetura Proposta Baseada em Filtros	34
Figura 3.2	Diagrama de Sequência para a Operação de Armazenamento dos Batimentos Cardíacos	37
Figura 3.3	Diagrama de Sequência para a Operação de Armazenamento da Pressão	38
Figura 3.4	Módulo de Envio de Mensagens SMS	39
Figura 3.5	Diagrama de Sequencia do Módulo Envia SMS	40
Figura 3.6	Módulo Móvel Utilizando a Arquitetura JME	41
Figura 3.7	Diagrama de Sequência do Módulo de Visualização Móvel	42
Figura 3.8	Arquitetura do Módulo TVDi	43
Figura 3.9	Diagrama de Sequencia Envia Alerta para a TVDi	43
Figura 4.1	Equipamentos Utilizados	47
Figura 4.2	Equipamentos Utilizados no Módulo de Coleta	48
Figura 4.3	Medidor de Pressão Arterial	49
Figura 4.4	Cenário de um Serviço de Coleta e Tratamento de Dados de Sensores.	50
Figura 4.5	Diagrama de Casos de Uso Coleta da Pressão Arterial	51
Figura 4.6	Diagrama de Classe Coleta da Pressão Arterial	52
Figura 4.7	Trecho do Código Fonte da Classe ParserDados	52
Figura 4.8	Documento XML e seu Respectivo DTD	53
Figura 4.9	Diagrama de Casos de Uso Coleta da Frequência Cardíaca	54
Figura 4.10	Diagrama de Classes e Pacotes Frequência Cardíaca	55
Figura 4.11	Trecho de Código Fonte do Método GetDados	55
Figura 4.12	Vetor com Dados Coletados do Oxímetro	56

Figura 4.13	Estrutura do Documento XML da Frequência Cardíaca	57
Figura 4.14	Diagrama de Caso de Uso do Módulo de Mensagens	58
Figura 4.15	Diagrama de Classe do Módulo de Mensagens	58
Figura 4.16	Trecho de Código do Módulo de Envio de Mensagens	59
Figura 4.17	Exemplo de um Alerta em Formato XML	60
Figura 4.18	Diagrama de Caso de Uso do Módulo TVDi	61
Figura 4.19	Diagrama de Classes do Módulo TVDi	62
Figura 4.20	Código Fonte do Método <i>keyPressed</i>	62
Figura 4.21	Diagrama de Estados da Aplicação	63
Figura 4.22	Diagrama de Classes Adaptado para Lua e NCL	64
Figura 4.23	Trecho de Código Fonte em LUA	64
Figura 4.24	Diagrama de Casos de Uso para a Aplicação Móvel	65
Figura 4.25	Diagrama de Classes para a Aplicação Móvel	66
Figura 4.26	Parte do Código Fonte da Classe XMLGetPressure	66
Figura 5.1	Tela de Apresentação de Alarme	68
Figura 5.2	Tela de Apresentação de Dados	69
Figura 5.3	Configuração dos Módulos XBee	70
Figura 5.4	Tela de Apresentação de Dados no STB	71
Figura 5.5	Tela de Apresentação de Dados no XletView	71

Lista de Tabelas

Tabela 2.1	Comparação entre os Trabalhos Relacionados	30
Tabela 3.1	Faixa Máxima da Frequência Cardíaca	35
Tabela 3.2	Classificação da Pressão Arterial em Adultos com Idade Igual ou Superior a 18 anos	35
Tabela 3.3	Cenário para a Coleta de Dados do Oxímetro	36
Tabela 3.4	Cenário para a Coleta de Dados da Pressão Arterial	38
Tabela 3.5	Cenário para Envio de SMS	40
Tabela 3.6	Cenário para a Apresentação de Dados no Celular	42
Tabela 3.7	Cenário para a Apresentação de Alertas na TVDi	44

Sumário

1	Introdução	16
1.1	Problema	18
1.2	Descrição dos Cenários Baseados no Problema	18
1.3	Motivação e Justificativa	19
1.4	Objetivo Geral	20
1.5	Objetivos Específicos	20
1.6	Metodologia	20
1.7	Organização do Trabalho	21
2	Fundamentos e Trabalhos Relacionados	23
2.1	Sistema Brasileiro de TV Digital	23
2.2	Gateway Residencial	27
2.3	Dispositivos de Hardware e Sensores	28
2.4	Trabalhos Relacionados	28
2.5	Conclusão	31
3	Modelo Arquitetural Proposto	33
3.1	Concepção da Solução	33
3.2	Cenários de Uso Aplicados à Arquitetura	35
3.2.1	Módulos de Coleta de Dados	36
3.2.2	Módulo de Envio de Mensagens SMS	37
3.2.3	Módulo de Visualização de Dados no Celular	39
3.2.4	Módulo de Apresentação de Alertas na TVDi	41
3.3	Conclusão	44
4	Implementação do Modelo Proposto de Acordo com os Cenários	46
4.1	Dispositivos e Tecnologias Utilizados na Implementação	46
4.1.1	Configuração e Instalação do Gateway Residencial (GR)	47
4.1.2	Configuração dos Dispositivos de Coleta de Dados	48
4.2	Implementação dos Cenários	49
4.2.1	Implementação do Cenário de Coleta de Dados da Pressão Arterial	50

4.2.2	Implementação do Cenário de Coleta de Dados da Frequência Cardíaca	53
4.2.3	Implementação do Cenário do Módulo de Envio de Mensagens . . .	57
4.2.4	Implementação do Cenário de Apresentação na TVDi e no Celular .	60
4.3	Conclusão	66
5	Testes e Resultados	68
5.1	Cenário Medir Pressão	68
5.1.1	Dispositivos de Hardware e Software Utilizados	69
5.1.2	Considerações sobre o protótipo	69
5.2	Cenário Medir Frequência Cardíaca	70
5.2.1	Dispositivos de Hardware e Software Utilizados	71
5.2.2	Considerações sobre o protótipo	72
5.3	Conclusão	72
6	Considerações Finais	74
6.1	Dificuldades Encontradas	74
6.2	Sugestões para Trabalhos Futuros	75
	Referências Bibliográficas	77
	Apêndice A – Publicações	80

Capítulo 1

Introdução

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), 98% dos brasileiros residentes em áreas urbanas tem acesso à televisão. Certamente esse percentual deve ser refletido para a TV Digital Interativa (TVDi) quando esta for implantada no Brasil em sua totalidade (IBGE, 2009). Aliado a esse fato, o IBGE também apresenta em suas pesquisas que a quantidade de idosos no Brasil representa mais de 19 milhões de pessoas com idade de 60 anos ou mais. Isto é 8,6% da população brasileira, que continua crescendo e deve chegar a 30 milhões nos próximos 20 anos, ou seja, quase 13% da população ao final deste período.

Além da população crescente de idosos no Brasil, também existe um fator preocupante na área de saúde que são as pessoas com diabetes e hipertensão, as quais necessitam de cuidados médicos e tratamentos especiais hoje fornecidos por programas de saúde dos Governos Federal e Estadual (CNDH, 2009).

A demanda por serviços especializados de saúde irá crescer na proporção do crescimento da população idosa, afetando o atendimento em hospitais e casas de saúde, visto que com o aumento da idade do cidadão aumenta também sua propensão a manifestar problemas de saúde.

Com o advento da convergência digital, o avanço das tecnologias de comunicação sem fio, o maior número de dispositivos embarcados e móveis e a implantação do *International System Digital Broadcasting Terrestrial Brazilian* (ISDB-TB) (FORUM-TVDI, 2009), surgem oportunidades para integrar essas tecnologias de forma a prover serviços de monitoramento e coleta de dados de usuários através de um sistema que integre dispositivos e sensores médicos convencionais, possibilitando gerar alarmes sobre o estado dos sensores conectados ao usuário e transmiti-los, por exemplo, a um telefone celular de um profissional de saúde.

Para um melhor entendimento do escopo deste trabalho serão apresentados os conceitos básicos de TVDi, dispositivos médicos convencionais, concentradores de dados

e ambientes inteligentes, e no decorrer dos capítulos seguintes dessa dissertação esses conceitos serão tratados mais especificamente.

A tecnologia de TVDi é composta por um conjunto de equipamentos de recepção e transmissão de sinais de áudio e vídeo com modulação digital, uma camada de software para integração entre aplicações e *hardware* chamado *middleware*, e um conjunto de aplicações. O avanço de novos algoritmos Codificadores e Decodificadores de Vídeo (CODEC), a transformação do sinal analógico para o digital e a interatividade através de um canal de retorno, possibilitam uma nova perspectiva ao telespectador que agora passa a interagir com o sistema de TVDi e sua programação (MORRIS, 2005).

A interatividade a ser proporcionada pela TVDi viabilizará o desenvolvimento de aplicativos que explorem essa tecnologia usando o canal de retorno a ser fornecido por provedores de serviços de telecomunicações, ainda em fase de definição no Brasil (FORUM-TVDI, 2009). Desta maneira, a interação entre telespectador e provedores de conteúdo será uma realidade cada vez mais próxima, o que possibilita o desenvolvimento de sistemas integrando dispositivos de redes, sensores e tecnologias sem fio como, por exemplo, dispositivos que usem a tecnologia *Bluetooth* ou *Zigbee*.

Outro dispositivo que também está se tornando comum nas residências é o *Gateway Residencial* (GR), conforme é apresentado em Maia (2009), Viana (2009) e em Lin e Yu-Hsiang-Sheng (2008), que consiste em um equipamento ligado a uma rede residencial com a função de coordenar a interconexão de outros dispositivos que se comunicam e fornecem serviços internos à residência, através do uso de sensores em eletrodomésticos e comunicação externa usando o compartilhamento da banda de *internet*.

Além da presença recente da TVDi e de dispositivos concentradores de dados no cotidiano das pessoas, existem também os ambientes inteligentes que tendem a ser cada vez mais presentes nas residências automatizadas, tornando possível o planejamento e a construção de ambientes que interajam com equipamentos eletrônicos de consumo e com sensores distribuídos na residência, através do uso das tecnologias como, por exemplo, Agentes Inteligentes (AI) e Ambient Intelligence (AmI), conforme é apresentado em Augusto, Carlos e Daniel (2008).

O conceito de AmI, introduzido em 2001 e modificado em 2002 e 2003 através de uma comissão europeia chamada de *Information Society Technologies Advisory Group* (ISTAG), conforme Augusto, Carlos e Daniel (2008) e Ducatel et al. (2001), proporciona ambientes sensíveis e responsivos à presença de pessoas dando ênfase na facilidade de utilização de serviços de apoio e capacitação de usuários e principalmente no apoio das interações humanas com interfaces inteligentes e intuitivas, embarcadas em todos os tipos

de objetos em ambientes capazes de reconhecer e reagir a presença de indivíduos diferentes. Possuem como características principais a invisibilidade, mobilidade, heterogeneidade, percepção, antecipação, interação com usuário, observação, aprendizado e adaptabilidade.

O uso de um GR para interconectar os dispositivos médicos com a TVDi carece de um estudo de viabilidade, pois os set-top box (STB) atuais comercializados no Brasil, ainda não possuem poder de processamento suficiente para gerenciar dispositivos em uma residência. Este trabalho propõe uma arquitetura para implementação de serviços de *healthcare* para coleta e transmissão de dados, usando a TVDi como interface com o usuário afim de integrar dispositivos médicos convencionais de baixo custo como por exemplo, medidores de pressão arterial, medidores de frequência cardíaca e um celular como canal de retorno da aplicação. Desta forma, pretende-se integrar esses dispositivos com a TVDi para viabilizar o desenvolvimento de aplicativos de *healthcare*.

1.1 Problema

A problemática objeto de estudo apresentado neste trabalho, consiste em verificar se é possível integrar dispositivos médicos convencionais existentes no mercado com a TV digital brasileira para prover serviços de *healthcare*.

1.2 Descrição dos Cenários Baseados no Problema

Com o objetivo de ilustrar o problema apresentado foram desenvolvido dois cenários que simulam a situação em que um usuário é monitorado através de um sistema de coleta de dados de um medidor de batimentos cardíacos e uma outra situação em que um sistema de alerta informa ao usuário através da TVDi o horário agendado para a medição de sua pressão arterial.

O primeiro cenário baseia-se em uma residência de um idoso que deseja monitorar sua pressão arterial e que já possui acompanhamento médico. Nesse cenário foi considerada uma residência em área urbana com acesso à *internet* e telefonia celular e uma rede doméstica integrando os dispositivos residenciais com a TVDi.

Cenário 1:

José, um senhor idoso que mora sozinho, tem problemas de pressão arterial e sempre esquece de fazer as aferições agendadas por seu médico. O médico costuma fazer uma visita por mês, para saber a situação de saúde de José, os filhos de José também costumam frequentar a casa do pai a fim de acompanhar o estado de saúde do mesmo.

Uma vez por mês, José faz seu check-up no hospital e recebe medicamentos para

controle da pressão. Para facilitar a vida de José, seus filhos instalaram um sistema de controle automatizado integrado a TVDi, a fim de alertar o pai para que o mesmo possa aferir a pressão arterial de acordo com o horário prescrito pelo médico na comodidade de sua residência.

O sistema além de processar as informações, é capaz de guardar o número de telefone, nome e endereço dos filhos de José, e de seu médico e enfermeiro. O sistema permite também que os filhos de José cadastrem seus celulares para receberem via rede de telefonia, mensagens de texto caso seu pai deixe de medir a pressão.

O segundo cenário, baseia-se em uma residência de um usuário que costuma se exercitar com frequência em sua residência usando uma esteira. Esse usuário faz exercícios de corrida e caminhada na esteira e coleta dados sobre a sua frequência cardíaca para melhorar sua performance durante os exercícios.

Cenário 2:

João costuma exercitar-se em casa usando uma esteira onde corre duas horas por dia três vezes por semana. João é cuidadoso com sua saúde e durante seus exercícios deseja acompanhar seu rendimento e o esforço que está se submetendo.

Ele decide adquirir um sistema de medição de batimentos cardíacos equipado com um oxímetro de pulso sem fio que lhe permite acompanhar sua frequência cardíaca através da TVDi enquanto ele assiste seu programa preferido.

De acordo com os cenários apresentados observa-se funcionalidades gerais para nortear o desenvolvimento da arquitetura e posteriormente a coleta de requisitos para protótipos do sistema apresentados nos capítulos seguintes dessa dissertação. A arquitetura proposta bem como os protótipos desenvolvidos foram objetos deste estudo e serão detalhados no decorrer do trabalho.

1.3 Motivação e Justificativa

Este trabalho tem como motivação a pesquisa para propor possíveis soluções para acompanhamento de sinais e medições fisiológicas de usuários usando dispositivos médicos convencionais com o intuito de diminuir os custos com equipamentos e fazer a integração com a TVDi.

A integração com as tecnologias móveis atuais do mercado e o desenvolvimento de um protótipo aplicado ao estudo de caso envolvendo o uso da TVDi, permitirá o intercâmbio de informações entre a residência de um usuário e seu médico através do uso de um telefone celular. Desta maneira, um paciente poderá ser acompanhado por profissionais de saúde à distância que receberão mensagens de texto sobre os sinais

emitidos por dispositivos conectados ao usuário.

A importância do trabalho justifica-se no uso da tecnologia para facilitar a vida de indivíduos que necessite de cuidados médicos, idosos que morem sozinhos e pessoas em geral que desejem monitorar sua pressão arterial e sua frequência cardíaca.

1.4 Objetivo Geral

O objetivo geral a ser atingido com esse trabalho é investigar e propor a concepção de um sistema integrado de informação com características específicas para execução de aplicações de *healthcare* compatível como o modelo brasileiro de TVDi.

1.5 Objetivos Específicos

Os objetivos específicos que direcionam esse trabalho são:

- Determinar as especificações mínimas de hardware de um *set-top box* para comportar um *gateway* de *healthcare*.
- Identificar e implementar em *software* a infraestrutura de um *gateway* de *healthcare*.
- Implementar a comunicação entre dispositivos médicos convencionais e a TVDi através de um *gateway* de *healthcare*.
- Investigar e propor um mecanismo apropriado para usar como canal de retorno nas aplicações de *healthcare* para a TVDi.
- Implementar uma aplicação de *healthcare* como estudo de caso para a TVDi brasileira.

1.6 Metodologia

A Figura 1.1 ilustra as etapas da metodologia. O trabalho foi dividido em quatro etapas iniciando com uma pesquisa bibliográfica na qual foram investigados os trabalhos relacionados envolvendo as tecnologias de TVDi e dispositivos médicos convencionais e as respectivas aplicações de *healthcare* existentes, fazendo uma revisão crítica da literatura em artigos nacionais e internacionais sobre o tema, finalizando com a catalogação dos trabalhos realizados.

Na segunda etapa, foram feitos experimentos de software, através do uso de equipamentos eletrônicos, celulares, sensores sem fio e STB disponíveis nos laboratórios

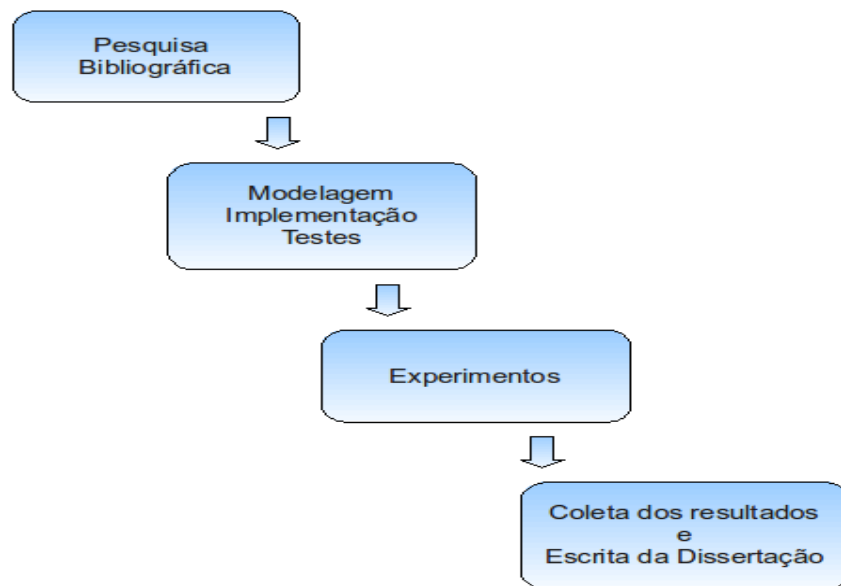


Figura 1.1: Etapas da Metodologia de Desenvolvimento do Trabalho.

Fonte: Do próprio autor.

do Centro de Tecnologia Eletrônica e Telecomunicações (CETELI). Para esta etapa a metodologia consistiu em: escolher aplicações já desenvolvidas apresentadas em artigos consolidados, repetir os experimentos catalogando e comparando os resultados a partir das conclusões observadas, refazer os experimentos introduzindo as modificações e contribuições necessárias.

Na terceira etapa foi modelada e implementada uma aplicação de *healthcare* integrada à TVDi através de um GR. As ferramentas e materiais para esta etapa foram um computador servidor de controle de versão para armazenamento e gerenciamento do código fonte, um computador configurado como GR, dois STB, sendo um com o *Middleware Brasileiro de TV Digital (GINGA)* instalado e o outro com o *middleware europeu Multimídia Home Platform (MHP)*, um oxímetro de pulso, um medidor automático de pressão arterial e um telefone celular.

Na etapa final os resultados foram apresentados em artigos submetidos a congressos relacionados ao tema, finalizando com a escrita da dissertação.

1.7 Organização do Trabalho

O presente trabalho está dividido nos capítulos listados a seguir. No capítulo um é apresentado a ambientação do trabalho conceitos iniciais sobre TVDi e GR, bem como um cenário para auxiliar o entendimento do problema, objetivos e a metodologia

utilizada.

O capítulo dois discorre sobre os fundamentos e trabalhos relacionados, onde os conceitos de GR, ISDB-TB, dispositivos médicos e sensores são apresentados e os trabalhos relacionados ao tema da dissertação são verificados e comparados.

No capítulo três, o modelo arquitetural da solução proposta é apresentado, enquanto que no capítulo quatro, a implementação do protótipo usando a arquitetura é demonstrado.

Finalizando a dissertação, os testes com STB real e emuladores são descritos no capítulo cinco e o capítulo seis apresenta as considerações finais e trabalhos futuros.

Capítulo 2

Fundamentos e Trabalhos Relacionados

Neste capítulo, são apresentados os conceitos que envolvem as tecnologias utilizadas para o desenvolvimento do trabalho e uma avaliação dos trabalhos relacionados ao tema é desenvolvida. As estruturas de uma aplicação para o Subsistema Procedural para a TV Digital Brasileira (GINGA-J) e Subsistema Declarativo para a TV Digital Brasileira (GINGA-NCL) são apresentadas e os conceitos de TVDi são explicados.

2.1 Sistema Brasileiro de TV Digital

A especificação do ISDB-TB foi baseada no padrão japonês *Integrated Services Digital Broadcasting Terrestrial* (ISDB-T) após a decisão do fórum brasileiro de tv digital terrestre, criado por um grupo liderado pela *Agência Nacional de Telecomunicações* (ANATEL) e pelo Ministério das Comunicações. O sistema brasileiro foi especificado com modificações no sistema de compressão de áudio e vídeo que agora passa a contar com a codificação H264 e HE-AAC v.2, com velocidade de até trinta quadros por segundo (ABNT, 2007).

A Figura 2.1 ilustra a especificação do *middleware* GINGA, na qual observa-se a integração de dois subsistemas, GINGA-J e GINGA-NCL. Entre eles está previsto uma ponte de ligação para possibilitar a troca de dados entre as aplicações desenvolvidas em *Nested Context Language* (NCL) e em Java TV. Essa ponte permite que aplicações escritas em NCL e Java TV se comuniquem, o que segundo Soares, Rodrigues e Moreno (2007) é uma novidade no *middleware* brasileiro.

Observando a Figura 2.1, nota-se que aplicações NCL e aplicações Xlet convivem lado a lado usando os recursos de codificação de vídeo e áudio através da camada de decodificação que por sua vez está integrada ao sistema Padrão de Codificação para Vídeo e Áudio Digital (MPEG-2) que acessa diretamente o *hardware* do STB.

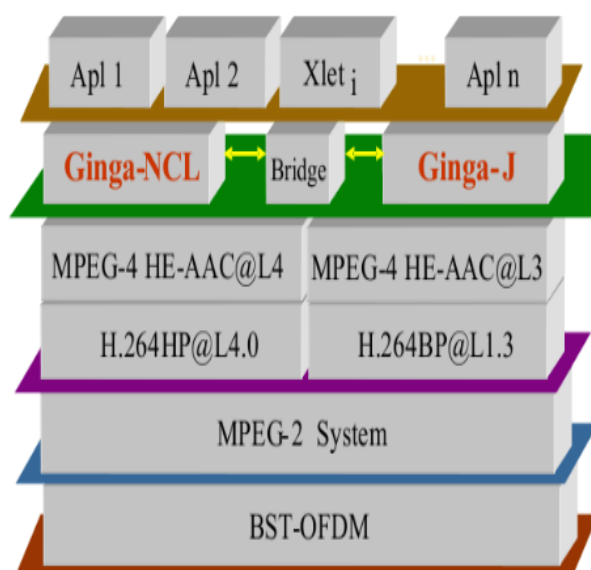


Figura 2.1: Arquitetura do *middleware* brasileiro GINGA.

Fonte: Adaptado de (SOARES; RODRIGUES; MORENO, 2007).

De acordo com Soares, Rodrigues e Moreno (2007) o ISDB-TB, suporta ambientes procedurais e declarativos através do *middleware* GINGA, que é subdividido em GINGA-J e GINGA-NCL, respectivamente ambiente procedural e ambiente declarativo.

O paradigma procedural embutido no *middleware* permite o uso da linguagem de programação Java através da especificação definida no GINGA-J. Desta maneira é possível escrever aplicações para a TVDi utilizando o paradigma de Programação Orientada a Objetos (POO) e integrar as *Application Programming Interface* (API) necessárias para comunicação de dados e desenvolvimento de interfaces gráficas mais robustas.

A API Java TV provê acesso às funcionalidades específicas para os receptores de TVDi que incluem dentre outros, acesso a serviços, seleção de serviços, informações sobre serviços, controle de gráficos, controle em alto nível da recepção do sinal (*broadcast*), difusão e gerência do ciclo de vida das aplicações Xlets. A API dá suporte às aplicações executadas na *Java Virtual Machine* (JVM)) de forma a viabilizar o tratamento do conteúdo interativo por parte da camada de aplicação.

Na Figura 2.2 é apresentada a integração dos componentes na arquitetura Java TV e o relacionamento entre as camadas. Nota-se que a camada de aplicação se integra com a API Java TV permitindo desta forma que outras classes Java sejam utilizadas no desenvolvimento da aplicação. A camada de *Real-time Operating System* (RTOS), inclui os *drivers* do receptor de TVDi e dá suporte à camada de *hardware*.

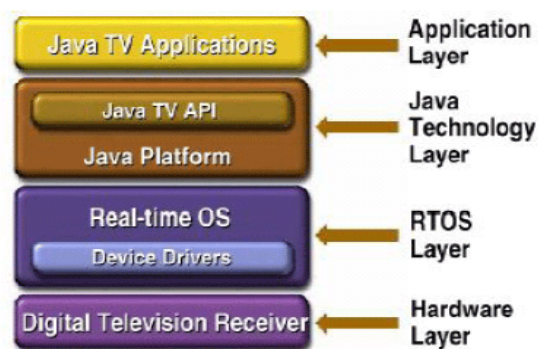


Figura 2.2: Arquitetura da API Java TV.

Fonte: (JAVA-TV, 2008).

Uma aplicação Xlet é composta por uma classe principal que implementa a interface Xlet e executa sobre a JVM do STB (MORRIS, 2005). Possui quatro estados no seu ciclo de vida a saber:

- *loaded* (carregado), estado que carrega o Xlet porém não o inicia. Esse estado é alcançado quando o *middleware* instancia a classe principal da aplicação e o carrega em memória. Em caso de falha o estado de *destroyed* (destruído) é chamado e a aplicação Xlet é destruída.

- *paused* (pausado), nesse estado o Xlet foi inicializado contudo não usa nenhum recurso compartilhado do STB. O estado é pausado de duas formas, quando o método *initXlet* do estado *loaded* é invocado ou quando o método *pauseXlet* é chamado a partir do estado *active* (ativo).

- *active* (ativo), o Xlet está em funcionamento, nesse caso o método *startXlet* é invocado a partir do estado *paused*.

- *destroyed* (destruído), o Xlet é destruído e libera todos os recursos que usou, ocorre quando uma exceção é lançada (*XletStateChangeException*), em qualquer transição de estados ou quando o método *destroyXlet* é invocado de qualquer dos estados. A figura 2.3 apresenta o ciclo de vida de uma aplicação Xlet.

A especificação do Ginga-J foi desenvolvida baseada na especificação Java TV em parceria com a *Sun Microsystem*, com a diferença que na especificação brasileira há o esforço para reduzir o pagamento de *royalties* tornando a plataforma simples e flexível e constituindo uma alternativa em relação ao *Globally Executable MHP* (GEM), utilizado nos padrões do *middleware* Europeu. Desta forma, espera-se reduzir os custos de produção de um STB com o padrão Brasileiro, e usar a tecnologia como veículo para promover a inclusão digital e social no país (ALMEIDA; GARCÍA, 2006).

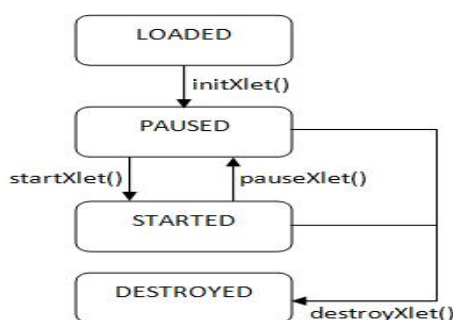


Figura 2.3: Ciclo de Vida de uma Aplicação Xlet

Fonte: Adaptado de (MORRIS, 2005).

O subsistema GINGA-NCL processa documentos no formato *Nested Context Language* NCL e é o ponto chave para a implementação de aplicações com conteúdo declarativo. Com ele é possível desenvolver marcações para controlar exibição de vídeos, bem como utilizar a linguagem de *script* Lua (IERUSALIMSKY, 2006) para promover a interatividade.

O conteúdo de um documento NCL possui uma estrutura que segue o padrão *Extensible Markup Language* (XML) e divide-se em cabeçalho e corpo. O cabeçalho guarda as bases da informação que especificam onde e como o conteúdo deve ser exibido (SANT'ANNA, 2009). O corpo descreve o conteúdo a ser exibido e em que momento isso deve ser feito.

A linguagem NCL controla os objetos de mídia de tipos diferentes e os relaciona. Por exemplo, um vídeo que encapsule uma informação interativa é descrito em um descritor de vídeo escrito em NCL que permite configurar o tamanho da tela de exibição enquanto um menu de opções pode ser visto no canto esquerdo da mesma tela. Isso é possível porque os descritores têm acesso à mídia, seus conectores e sua região de tela. Desta maneira, pode-se inferir quando e como a mídia é apresentada. Um exemplo de um arquivo NCL simples é apresentado na Figura 2.4

O arquivo NCL apresentado na Figura 2.4 é parte de uma configuração inicial para redimensionamento de telas de uma aplicação de mensagens instantâneas e possui as configurações de região base de um arquivo de vídeo inicialmente em tela cheia, identificados pelo nó *regionBase* e sub-nó *regionId*. Em seguida o nó *descriptorBase* aponta para região e por fim o nó *port*, chama o componente Lua para execução das configurações enquanto que o nó *media* indica o caminho para o arquivo de vídeo.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="chat_tela" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <regionBase>
      <region id="rgVideoFullscreen" zIndex="1"/>
      <region id="rgLua" width="100%" height="100%" right="0" top="0" zIndex="10" />
    </regionBase>

    <descriptorBase>
      <descriptor id="dVideoFullscreen" region="rgVideoFullscreen"/>
      <descriptor id="dLua" region="rgLua" focusIndex="1" />
    </descriptorBase>
  </head>

  <body>
    <port id="pLua" component="lua" />
    <media id="lua" src="main.lua" descriptor="dLua" />
    <media id="videoAbertura" src="img/background.jpg" descriptor="dVideoFullscreen" />
  </body>
</ncl>

```

Figura 2.4: Exemplo de um Arquivo NCL

Fonte: Do próprio autor.

2.2 Gateway Residencial

Um *gateway* residencial é um dispositivo de *hardware* e *software* que faz a ponte entre dispositivos conectados em rede e os ambientes externos. Maia (2009) e Viana (2009), apresentam um *gateway* residencial (GR) como um concentrador e gerenciador de dispositivos usando OSGi, aplicado a um estudo de caso de automação residencial. Em Lin e Yu-Hsiang-Sheng (2008) um GR foi utilizado para controle de sensores de glicose em um ambiente integrado com o sistema de TVDi europeu. Esse GR faz o processamento das informações e as armazena para consultas posteriores. A Figura 2.5 ilustra um GR conectando os dispositivos em uma rede residencial.

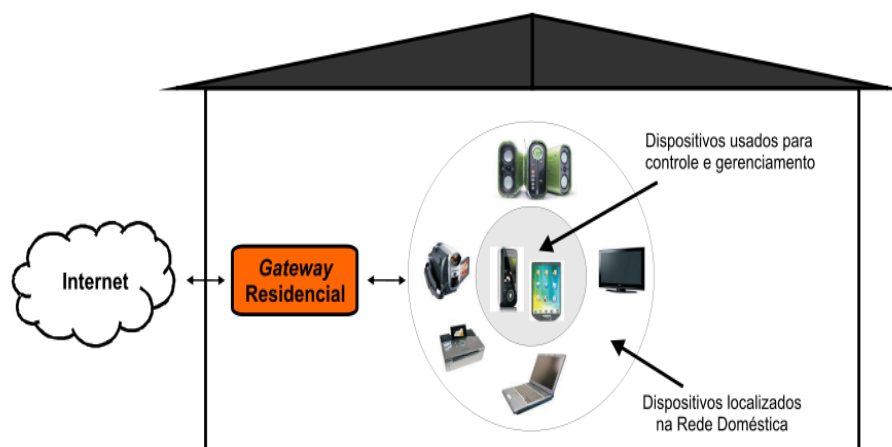


Figura 2.5: Visão Geral de uma Infra-estrutura de Redes Residenciais Usando um GR

Fonte: (MAIA, 2009).

No Brasil, os dispositivos mais comumente conhecidos, e que se aproximam de um GR, são os roteadores de *internet* banda larga, que permitem conectar uma rede residencial às redes externas e à internet, fornecendo serviços de compartilhamento de dados em uma rede residencial e fora dela.

Nessa dissertação, o GR é tratado como um concentrador dos dados que são coletados dos sensores e dispositivos médicos, e tem como tarefa tratar esses dados afim de enviar alarmes para um dispositivo móvel e uma TVDi compatível com o modelo brasileiro de TV Digital.

2.3 Dispositivos de Hardware e Sensores

Atualmente, existem uma gama de dispositivos médicos no mercado que podem ser integrados através de sistemas para auxiliar no diagnóstico e acompanhamento de doenças crônicas. Dentre esses dispositivos podem ser citados os medidores automáticos de pressão arterial e os oxímetros de pulso.

A integração desses dispositivos em uma rede residencial para tratamento e prevenção de doenças, já é bastante utilizado nos países desenvolvidos, como os Estados Unidos e países da União Europeia. Trabalhos como o de Augusto, Carlos e Daniel (2008), após serem apresentados à comunidade científica transformam-se em produtos que podem ser encontrados no mercado daqueles países.

Os dispositivos mais comuns utilizados em aplicações de *healthcare*, são os medidores de glicose, bombas de insulina automáticas, medidores de pressão arterial, oxímetros, balanças eletrônicas, dentre outros que podem ser encontrados no mercado. Para um GR geralmente se usa um dispositivo de *hardware* com um sistema embarcado com funções específicas e bem definidas. Contudo, alguns experimentos usando a arquitetura X86, também são encontrados na literatura, como por exemplo o apresentado em Lin e Yu-Hsiang-Sheng (2008).

2.4 Trabalhos Relacionados

Os trabalhos relacionados ao tema da dissertação foram selecionados a partir da pesquisa bibliográfica nos periódicos das áreas de informática e saúde, e foram analisados para identificar como a tecnologia está sendo aplicada para fornecer serviços de *healthcare*. Desse estudo nasceu a proposta descrita nessa dissertação que resolverá os problemas relatados nos trabalhos pesquisados. Por fim, uma tabela contendo as principais funcionalidades dos protótipos apresentados na bibliografia foram comparados

com a proposta dessa dissertação.

Em Corchado, Bajo e Abraham (2008) uma implementação de *healthcare* envolvendo idosos foi desenvolvida através de um sistema de monitoramento para uma casa de saúde. Este sistema possibilitou o acompanhamento de pacientes com o mau de Alzheimer, através de etiquetas *Radio Frequency Identification* (RFID) espalhadas pela residência e no pulso de cada idoso. Assim, quando o idoso se aproximava de alguma porta ou janela o sistema lia o dispositivo RFID no seu pulso e enviava o sinal ao servidor de localização instalado na administração da casa. O sistema foi integrado em uma rede sem fios, que permitia que enfermeiras e médicos tivessem acesso à localização do paciente usando um *Personal Digital Assistant* (PDA).

Neste caso os autores identificavam os pacientes em diversas áreas da casa como por exemplo, cozinha, corredores, banheiros e quartos. As áreas de riscos foram identificadas e o acesso a determinadas salas foi controlado, evitando que os idosos saíssem da casa de saúde.

As tecnologias utilizadas foram uma rede residencial sem fios coordenada por um servidor de redes através de uma central de processamento que monitorava as pulseiras RFID, PDAs distribuídos entre os cuidadores de idosos, e as etiquetas RFID instaladas nos cômodos da residência.

Outro trabalho apresentado em Lin e Yu-Hsiang-Sheng (2008), provê um ambiente ubíquo de *healthcare* através da integração de uma rede de sensores ZigBee o *framework Open Services Gateway Initiative* (OSGI) e a TV digital, usando o padrão Europeu MHP. Os autores apresentaram uma arquitetura e definiram um protocolo de comunicação para converter dispositivos ZigBee em UPnP, através do mapeamento de endereços de *Internet Protocol* (IP), diretamente na porta de comunicação desses dispositivos. Assim um dispositivo de coleta de dados, como por exemplo, um coletor de glicose, pôde ser usado para coletar e armazenar os dados do paciente em um banco de dados, para posteriormente ser acessado pela WEB. Os autores investigaram nesse estudo como integrar um ambiente em OSGI com o ambiente de TV digital padrão MHP. Nesse trabalho a quantidade de tecnologias e camadas envolvidas e os serviços disponibilizados dependem do OSGI que nesse caso foi integrado com o MHP.

No trabalho de Valero e L.Vadillo (2007), foi apresentado um *framework* para serviços inteligentes de *healthcare* onde um sensor de detecção de quedas envia a informação para um sistema de decisão baseado em ontologias e WEB semântica. Dependendo da base de regras analisada pelo sistema ele dispara um alarme na casa usando uma sirene e em seguida aciona o serviço de ambulância e equipes de saúde locais

para o atendimento. A abordagem científica desse estudo, foi a de inteligência artificial, usando uma base de regras baseadas em ontologias, através de uma camada de razão que foi desenvolvida com a coleta de dados e questionários de especialistas da área de saúde. Em seguida a base foi inserida no sistema Protégé (PROTÉGÉ, 2009), através de uma interface gráfica que gerava os arquivos *Web Ontology Language* (OWL) para serem interpretados pela camada de inteligência. Nessa abordagem, os autores ainda necessita de profissionais para atualizar constantemente a base de regras o que é um ponto fraco no sistema pois a interferência humana na digitação e geração das regras, podem introduzir erros e a camada de decisão pode trabalhar com dados inconsistentes.

Em Pekhteryev, H. e Challa (2008), é apresentado um *framework* para um sistema de *healthcare* usando *smarts phones* que possuem sensores sem fios conectados ao corpo do paciente, permitindo que seus sinais vitais sejam monitorados através de uma *Body Sensor Network* (BSN). O sistema detecta os sinais vitais do paciente e os envia para um servidor que os processa e em seguida os envia para um centro de saúde especializado. O objetivo desse trabalho foi garantir a segurança nas transmissões usando uma rede BSN. Essa abordagem, permite que através da implementação da arquitetura, seja possível, acompanhar pacientes hipertensos na própria residência, e monitorar eventos que ocorram nos dispositivos e nos sensores através de um celular ou pela WEB. Nesse trabalho os autores usam um concentrador de dados que recebe as conexões dos dispositivos móveis e após processamento, enviam os dados para o destino. Porém, se os dados fossem enviados do PDA direto para base de dados, usando um arquivo comum, como por exemplo um arquivo XML, haveria uma diminuição no tráfego de rede e como consequência uma melhoria na conexão entre a rede BSN e o servidor.

Tabela 2.1: Comparação entre os Trabalhos Relacionados

Features	Corchardo	Lin	Valero	Pekhteryev	Dissertação
Dispositivos sem fio	sim	sim	sim	sim	sim
Alarm via SMS	não	não	não	não	sim
Acesso WEB	não	não	não	não	sim
Alarm via DTV	não	não	não	não	sim
Decisão	não	não	sim	não	sim
Segurança	não	não	não	sim	não

Na Tabela 1 é mostrada a comparação entre as características dos trabalhos relacionados e o presente trabalho. Basicamente a maioria dos trabalhos relacionados utiliza um GR como concentrador e coletor de dados, diferenciando-se nos padrões e

estilos de arquitetura. O trabalho que mais se aproxima dessa dissertação é o descrito em Valero et al. (2007), que tem uma camada de decisão implementada com ontologias e WEB semântica. As aplicações de *healthcare* integradas à televisão desenvolvidas no Brasil ainda são poucas, comparadas com a Europa e os Estados Unidos, entretanto, existem aplicações piloto para testar a interatividade da TVDi voltadas para a prevenção. É o caso das aplicações NCL-Peso Ideal e NCL-Viva Bem Saúde, ambas desenvolvidas pelo grupo de NCL da PUC-RIO.

A aplicação para TVDi apresentada em (BECKER; SOARES, 2008b), permite que o telespectador informe sua altura e receba a informação de qual é seu peso ideal com base no Índice de Massa Corporal (IMC). A altura é informada através dos números do controle remoto, sem vírgulas ou ponto. A aplicação é desenvolvida em NCL e a interface de dados e processamento são implementadas usando a linguagem de *script* Lua.

Na aplicação apresentada em (BECKER; SOARES, 2008a), é permitido que o telespectador escolha, através do controle remoto, a refeição que ele julga mais saudável entre quatro opções diferentes apresentadas. Após a escolha da refeição, a TVDi apresenta ao telespectador uma mensagem de texto informando se o prato escolhido faz parte de uma alimentação saudável ou não. A aplicação é implementada em sua totalidade na linguagem NCL.

Essas aplicações foram desenvolvidas para testar a interatividade local através do controle remoto, com o objetivo de apresentar as potencialidades das linguagens NCL e Lua, e divulgar o desenvolvimento de software para a TVDi. As aplicações com o código fonte estão disponíveis no Portal do Software Público brasileiro. (PÚBLICO, 2009)

O estudo apresentado neste capítulo, serviu para direcionar o autor para a escolha das tecnologias e dispositivos que serão utilizados na arquitetura e implementação do trabalho. A partir daí, decidiu-se pelo uso de um GR para processamento das informações. Estas serão coletadas por um oxímetro e por um medidor de pressão arterial, usando uma rede residencial e um celular para envio de SMS, através da aplicação embarcada no GR.

2.5 Conclusão

Nesse capítulo, foram apresentados os conceitos e as tecnologias utilizadas, bem como os trabalhos relacionados ao tema da dissertação focando nas funcionalidades de cada trabalho. A partir da verificação desses trabalhos, uma tabela relacionando itens comuns foi discutida e os pontos divergentes entre os trabalhos foram identificados. A arquitetura do *middleware* brasileiro de TVDi e suas sub-divisões foi apresentada e explicada, e os dispositivos coletores de dados encontrados no mercado foram citados.

No próximo capítulo será apresentado o Modelo Arquitetural proposto nessa dissertação e a contextualização dos cenários que foram apresentados no Capítulo 1.

Capítulo 3

Modelo Arquitetural Proposto

Neste capítulo será apresentada a arquitetura para implementação de serviços de *healthcare* e discutido o modelo para posterior implementação de protótipos. A concepção da arquitetura será descrita em um nível mais alto e em seguida detalhada em módulos.

3.1 Concepção da Solução

O modelo arquitetural usado neste trabalho é baseado na arquitetura de *pipes* e filtros apresentada em Mendes (2002), a qual consiste em tratar os dados aplicando filtros em vários níveis, transformando o dado bruto coletado dos dispositivos em informações legíveis para o usuário.

Segundo Mendes (2002), *pipes* e filtros consideram a existência de uma rede pela qual fluem dados de uma extremidade de origem à de destino, e o fluxo de dados sofre transformações através dos filtros. Nesta dissertação, *pipes* e filtros são utilizados de forma unidirecional, conduzindo e tratando os dados entre a fonte (sensores) e o destino (GR e TVDi).

A partir da concepção do modelo arquitetural apresentado na Figura 3.1, observa-se o uso de dois filtros de dados no sistema de processamento que trabalham na transformação dos dados recebidos através de protocolos de comunicação e um *parser*. Após o trabalho dos filtros um documento comum é gerado, validado e armazenado na base de dados.

Observa-se também que são previstos quatro módulos, a saber: módulo de dispositivos, módulo de usuários, módulo de processamento e módulo de dados. Cada módulo é responsável por funcionalidades específicas e bem definidas e se comunica através de um sub-módulo existente dentro da unidade de processamento que possui os filtros de protocolos de comunicação e *parser* de dados.

O fluxo dos dados que vai dos sensores até a base de dados, passa primeiramente

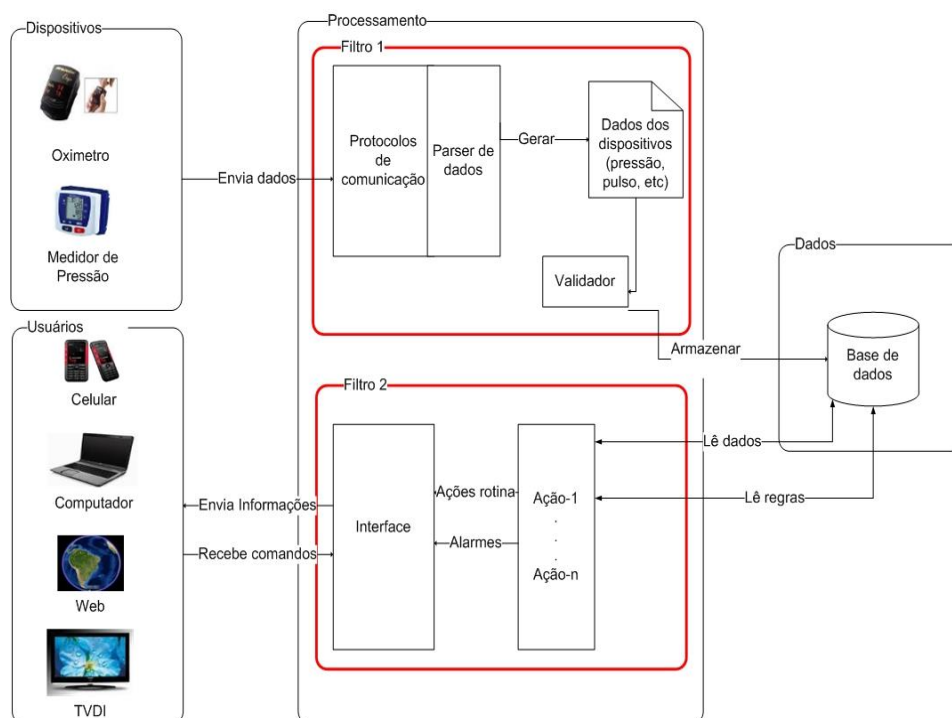


Figura 3.1: Arquitetura Proposta Baseada em Filtros

Fonte: Do próprio autor.

por um filtro de protocolos de comunicação, que é processado por um sub-módulo de *parser* de dados que, em seguida, gera o documento contendo os dados dos sensores. Esses dados são validados, através do sub-módulo validador e são armazenados na base de dados.

A partir da geração de um documento de dados que pode ser acessado por todos os módulos, e a confirmação do validador quanto a origem e formatação do documento, o segundo filtro de dados através do módulo de ação, consulta o documento validado para verificar se os valores de pressão e frequência cardíaca estão de acordo com os estipulados pelo profissional de saúde.

Usando uma interface com os dispositivos que apresentam os dados para o usuário, tais como, celulares, computadores conectados a WEB, e STB com o padrão de TVDi brasileiro, o sub-sistema de mensagens executa as ações de envio de mensagem SMS.

O módulo de ação necessita verificar uma tabela de regras para envio de mensagens para a TVDi e para o celular, portanto tomando por base os trabalhos divulgados por Camarda e Pinto (2005), Branco, Vianna e Lima (2004) e em Chobanian et al. (2003) um documento contendo essas regras deve ser desenvolvido e armazenado na base de dados. Dois exemplos desses dados podem ser vistos na Tabela 3.1 e na Tabela 3.2 que apresentam a faixa máxima da frequência cardíaca esperada para um indivíduo em exercício físico e

Tabela 3.1: Faixa Máxima da Frequencia Cardíaca**Fonte:** (BRANCO; VIANNA; LIMA, 2004)

Idade	Freq. Máxima	Freq.Ideal 75%	ZonaAlvo entre 70% e 80%
20	200	150	140 a 170
25	195	146	137 a 166
30	190	142	133 a 162
35	185	139	130 a 157
40	180	135	126 a 153
45	175	131	123 a 149
50	170	127	119 a 145
60	160	120	112 a 136
65	155	116	109 a 132
70	150	116	105 a 128

Tabela 3.2: Classificação da Pressão Arterial em Adultos com Idade Igual ou Superior a 18 anos**Fonte:** (CHOBANIAN et al., 2003)

Sistólica	Diastólica	Categoria
menor que 130	menor que 85	Normal
130-139	85-89	Normal alta
140-159	90-99	hipertensão branda(estágio 1)
160-179	100-119	hipertensão moderada(estágio 2)
180-209	110-119	hipertensão grave(estágio 3)
maior ou igual 210	maior ou igual 120	hipertensão muito grave(estágio 4)

a classificação da pressão arterial em adultos maiores de 18 anos.

Os principais desafios para implementar a solução encontrada são a integração da TVDi com os dispositivos médicos convencionais encontrados no mercado, a sincronização entre os dispositivos durante o acesso à base de dados comum, e a implementação das regras para confecção das mensagens para serem disponibilizadas nos dispositivos de apresentação. Isso se dá pelo fato da não padronização dos dados enviados pelos dispositivos, pois cada fabricante usa padrões e protocolos próprios para encapsulamento e envio dos dados dos sensores e dispositivos.

Nos tópicos seguintes, a arquitetura dos módulos de mensagem e os dispositivos de apresentação de dados na TVDi e no celular serão apresentadas usando cenários de uso.

3.2 Cenários de Uso Aplicados à Arquitetura

Os requisitos de um sistema são definidos na fase inicial de um processo de desenvolvimento e durante o processo de elicitação, o arquiteto faz uso de informações

Tabela 3.3: Cenário para a Coleta de Dados do Oxímetro

Nome do Cenário	Coleta de Dados do Oxímetro
Resumo	Os sensores devem ser ligados e conectados ao usuário
Fluxo normal de eventos	1. Inicia-se a coleta de batimentos 2. O sensor envia os dados via rede para o GR 3. O <i>parser</i> processa e disponibiliza os dados em um documento comum 4. O validador verifica a consistência dos dados e os armazena na base de dados
Fluxo alternativo de eventos	Nenhum
Exceções no fluxo normal de eventos	No passo 4, caso o validador não valide os dados eles são descartados
Estado anterior ao inicio do cenário	Aguardando leitura do dispositivo
Estado posterior ao cenário	Dados armazenados na base
Estímulo	Usuário liga os sensores
Pré-condições	Sistema encontra-se operacional
Pós-condições	Sistema armazena os dados dos sensores
Cenário desenvolvido por	Vandermi Silva

do domínio do problema buscando a identificação de peculiaridades do sistema a ser desenvolvido (SOMMERVILLE, 2007).

A elicitação dos requisitos desse trabalho usou cenários de uso para representação do fluxo de dados dos dispositivos e interação entre usuário e sistema. Um exemplo de cenário para auxílio à especificação da arquitetura baseada na coleta de dados do oxímetro de pulso pode ser visto na Tabela 3.3, apresentada na seção de módulos de coleta de dados.

3.2.1 Módulos de Coleta de Dados

A partir da observação da tabela 3.3, nota-se que o cenário apresentado, é constituído de uma parte do diagrama de sequência apresentado na Figura 3.2, e ilustra a comunicação entre os módulos de protocolo, *parser* e validador de dados.

Através da tabela nota-se a descrição do cenário onde, via fluxo normal de eventos, a coleta de dados é inicializada com o envio dos dados para o GR e caso haja uma exceção o validador descarta os dados e lê novamente a porta de comunicação.

O diagrama de sequência apresentado na Figura 3.2, ilustra a comunicação entre os módulos de Sensores, Processamento, Validador e Base de Dados. O módulo Sensor envia os dados para o processamento usando a funcionalidade `enviaDados`, o módulo de processamento processa e transforma os dados e em seguida gera o documento para ser validado pelo módulo validador. Se o arquivo for válido ele é salvo na base de dados.

No diagrama apresentado as operações de transformação e salvamento dos dados

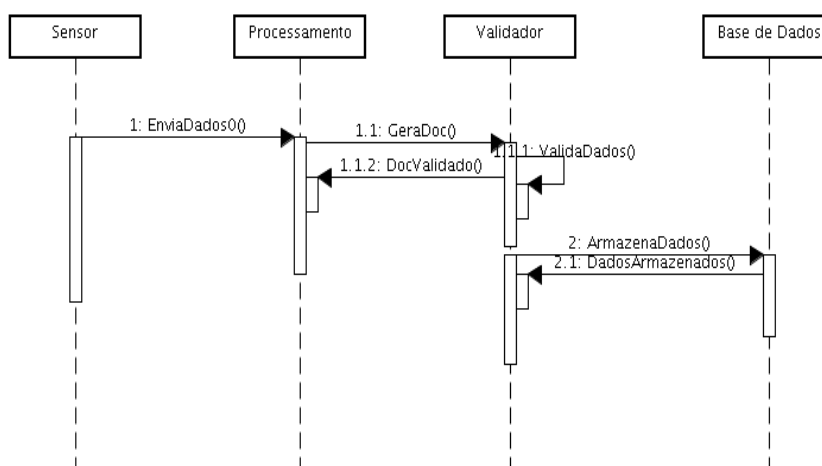


Figura 3.2: Diagrama de Sequência para a Operação de Armazenamento dos Batimentos Cardíacos

Fonte: Do próprio autor.

são feitas usando os métodos `geraDoc`, `validaDados` e `armazenaDados` que após sua execução, notificam os módulos que os chamaram, através de mensagens trocadas entre eles.

Na Figura 3.3 é ilustrada a comunicação entre os módulos de coleta, tratamento e armazenamento de dados do dispositivo coletor de pressão arterial. O diagrama de sequência da Figura 3.3, mostra a comunicação entre os módulos.

A tabela mostrada em 3.4 trata do cenário para medição de pressão arterial, onde um medidor de pressão automático coleta os dados do usuário e os envia para o GR. Durante o fluxo normal de eventos os dados são coletados, validados e gravados na base de dados. Caso os dados não sejam validados, uma mensagem é enviada para o usuário através de uma interface com a TVDi, solicitando que o mesmo refaça o processo de coleta da pressão arterial.

3.2.2 Módulo de Envio de Mensagens SMS

O módulo de envio de mensagem, usa a biblioteca `SMSLib` desenvolvida em Java e apresentada em (SMSLIB, 2008). Essa biblioteca possui os métodos para envio e recebimento de mensagens de texto e permite que um celular seja conectado em um computador através da porta *Universal Serial Bus* (USB). A arquitetura do módulo de mensagens *Short Message System* (SMS) é ilustrada na Figura 3.4, e executa sobre a camada da JVM instalada no GR. Desta forma, uma mensagem de texto pode ser disparada para celulares cadastrados no GR, usando a infra-estrutura da rede *Global System for Mobile Communications* (GSM) e seus protocolos de comunicação.

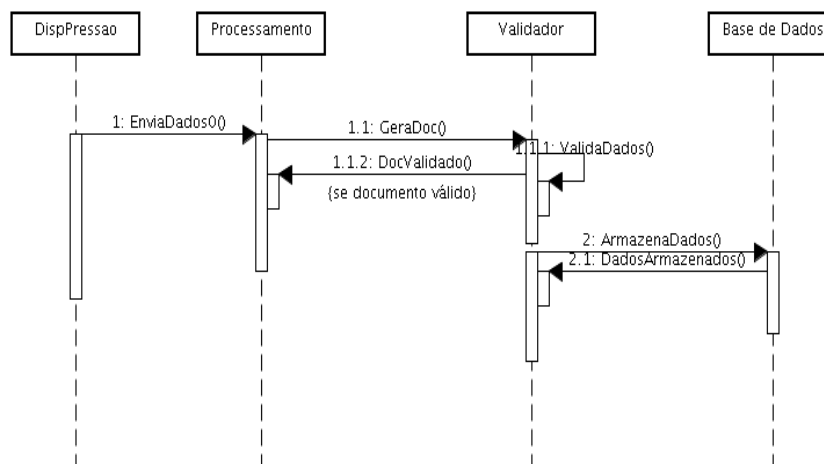


Figura 3.3: Diagrama de Sequência para a Operação de Armazenamento da Pressão

Fonte: Do próprio autor.

Tabela 3.4: Cenário para a Coleta de Dados da Pressão Arterial

Nome do Cenário	Coleta de Dados de Pressão Arterial
Resumo	O dispositivo medidor deve ser ligado e conectados ao usuário
Fluxo normal de eventos	<ol style="list-style-type: none"> 1. Inicia-se a coleta dos valores de pressão 2. O dispositivo envia os dados via rede para o GR 3. O <i>parser</i> processa e disponibiliza os dados em um documento comum 4. O validador verifica a consistência dos dados e os armazena na base de dados
Fluxo alternativo de eventos	Nenhum
Exceções no fluxo normal de eventos	No passo 4, caso o validador não valide os dados eles são descartados e uma mensagem é disponibilizada informando o usuário para medir novamente a pressão arterial
Estado anterior ao início do cenário	Aguardando leitura do medidor
Estado posterior ao cenário	Dados armazenados na base
Estímulo	Usuário liga o medidor de pressão
Pré-condições	Sistema encontra-se operacional
Pós-condições	Sistema armazena os dados do medidor de pressão
Cenário desenvolvido por	Vandermi Silva

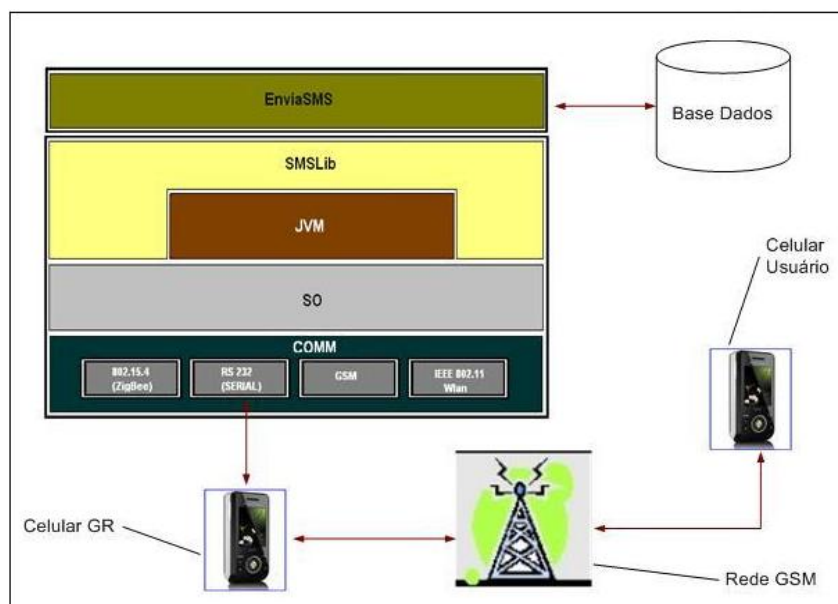


Figura 3.4: Módulo de Envio de Mensagens SMS

Fonte: Do próprio autor.

A mensagem é lida da base de dados, processada pelo módulo *EnviaSMS*, em seguida uma porta de comunicação entre o GR e o celular é estabelecida através da biblioteca *SMSLib*, então o celular do GR usando a rede GSM envia a mensagem para o celular de destino cadastrado no sistema. O módulo de mensagens SMS apresentado, é utilizado para o envio de mensagens de alerta da pressão arterial e da frequência cardíaca.

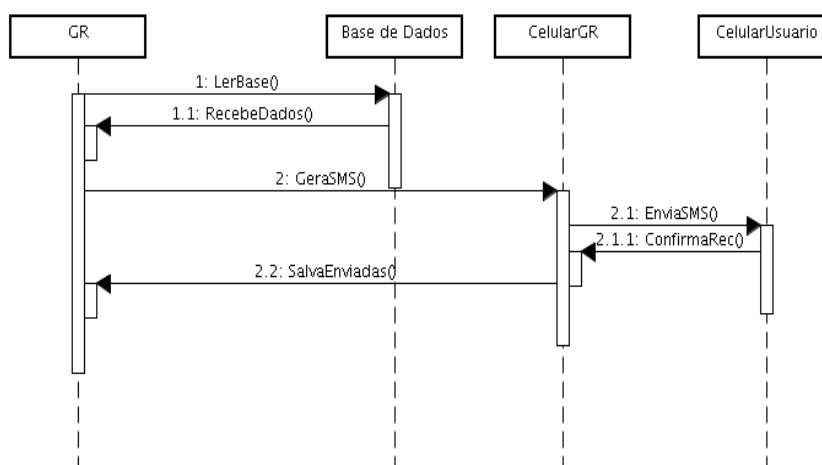
Observando a Figura 3.4, nota-se que seria possível usar outra implementação da biblioteca *SMSLib* que interage diretamente com o sistema operacional, pois no projeto da biblioteca existe um módulo para acesso nativo. Contudo nessa dissertação foi utilizada a biblioteca desenvolvida em Java por questões de portabilidade e facilidade para integração dos módulos do GR. A Tabela 3.5 lista o cenário de uso do módulo de envio de mensagens SMS e o diagrama de sequência da Figura 3.5, ilustra a comunicação entre os módulos.

3.2.3 Módulo de Visualização de Dados no Celular

O módulo de visualização de dados no celular, foi baseado na arquitetura *Java Micro Edition* (JME) vista em (HARKEY; APPAJODU; LARKIN, 2002). A partir da camada de aplicação têm-se acesso às camadas de *profile* e *Mobile Information Device Profile* (MIDP) que permite o acesso direto à camada *Connected Limited Device Configuration* (CLDC), responsável pelas configurações de dispositivos, acessíveis à *Kilobyte Virtual Machine* (KVM), uma máquina virtual Java específica para dispositivos pequenos.

Tabela 3.5: Cenário para Envio de SMS

Nome do Cenário	Cenário para Envio de SMS
Resumo	O celular do GR deve está ligado e com créditos
Fluxo normal de eventos	<ol style="list-style-type: none"> 1. Módulo de envio de SMS cria uma mensagem 2. Módulo SMS abre a conexão com o celular do GR 3. O celular processa e envia os dados via GSM 4. O celular do usuário recebe a mensagem
Fluxo alternativo de eventos	Nenhum
Exceções no fluxo normal de eventos	No passo 3, caso o celular do GR não estiver ligado ou não possuir créditos uma mensagem é disponibilizada informando o usuário que não foi possível enviar a mensagem
Estado anterior ao início do cenário	Aguardando criação da mensagem
Estado posterior ao cenário	Mensagem recebida no destino
Estímulo	Usuário verifica se celular está ligado
Pré-condições	Celular carregado ligado e com crédito
Pós-condições	Mensagem recebida no destino
Cenário desenvolvido por	Vandermi Silva

**Figura 3.5:** Diagrama de Sequencia do Módulo Envia SMS

Fonte: Do próprio autor.

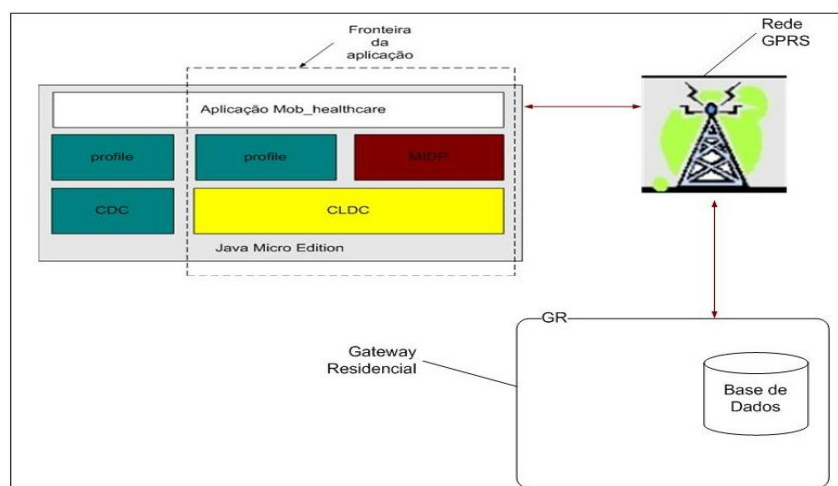


Figura 3.6: Módulo Móvel Utilizando a Arquitetura JME

Fonte: Do próprio autor.

A camada de aplicação limita-se ao uso da MIDP com CLDC, para diminuir os custos de memória e processamento, possibilitando que celulares de baixo custo sejam usados na implementação dos protótipos.

A Figura 3.6, ilustra desde a aquisição dos dados através de uma conexão *General Packet Radio Service* (GPRS) que acessa a base de dados do GR, até a apresentação dos dados no *display* do celular. Na Tabela 3.6, o cenário desenvolvido para o sistema de apresentação de mensagens no celular é ilustrado e na Figura 3.7 pode ser visto o diagrama de sequência que mostra a comunicação entre o celular o GR e a base de dados. Percebe-se que a comunicação ocorrida neste cenário parte do celular, que solicita acesso aos sensores e têm como resposta o valor daquele sensor no *display*.

3.2.4 Módulo de Apresentação de Alertas na TVDi

Nesta seção é apresentada o módulo de alertas baseado na arquitetura da TVDi brasileira, conforme a Figura 3.8 e em seguida o cenário de uso da arquitetura é mostrado na Tabela 3.7.

A partir da Figura 3.8, nota-se que o módulo de TVDi usa as especificações do *middleware* brasileiro apresentado em (SOARES; RODRIGUES; MORENO, 2007) e as aplicações podem ser implementadas em Java ou em NCL-LUA usando a camada de aplicação, enquanto a comunicação com o GR é feita através da camada de redes, já implementada nos STB encontrados no mercado.

O módulo de alerta recebe os dados via rede através da leitura dos dados da base de dados do GR, e em seguida mostra na tela da TVDi. A Figura 3.9 ilustra a comunicação

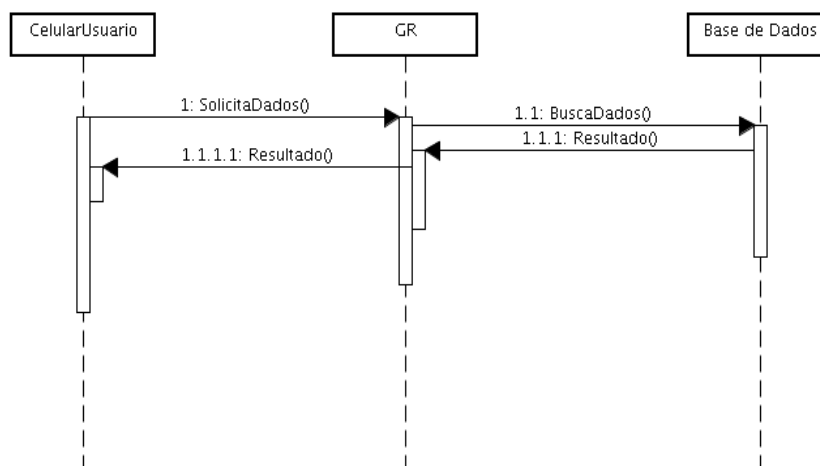


Figura 3.7: Diagrama de Sequência do Módulo de Visualização Móvel

Fonte: Do próprio autor.

Tabela 3.6: Cenário para a Apresentação de Dados no Celular

Nome do Cenário	Apresentação de Dados no Celular
Resumo	O celular deve está ligado e a aplicação móvel inicializada
Fluxo normal de eventos	<ol style="list-style-type: none"> 1. Usuário carrega a aplicação e solicita acesso aos sensores 2. O dispositivo envia a solicitação via rede GPRS para o GR 3. O GR devolve os valores do sensor via rede 4. O sistema móvel apresenta os dados no display
Fluxo alternativo de eventos	Nenhum
Exceções no fluxo normal de eventos	No passo 2, caso o usuário não consiga conectar na rede o display apresenta uma mensagem solicitando que o usuário tente outra vez
Estado anterior ao inicio do cenário	Sistema pronto para solicitar acesso
Estado posterior ao cenário	Dados apresentados no display
Estímulo	Usuário inicializa o sistema móvel
Pré-condições	Sistema encontra-se operacional
Pós-condições	Sistema mostra dados no display do celular
Cenário desenvolvido por	Vandermi Silva

entre os módulos participantes dessa funcionalidade. A partir dela é possível observar que na operação de envio de alertas, o GR primeiramente consulta a base de dados para verificar se existe um alerta, em seguida envia o alerta para o módulo de TVDi que o mostra na tela.

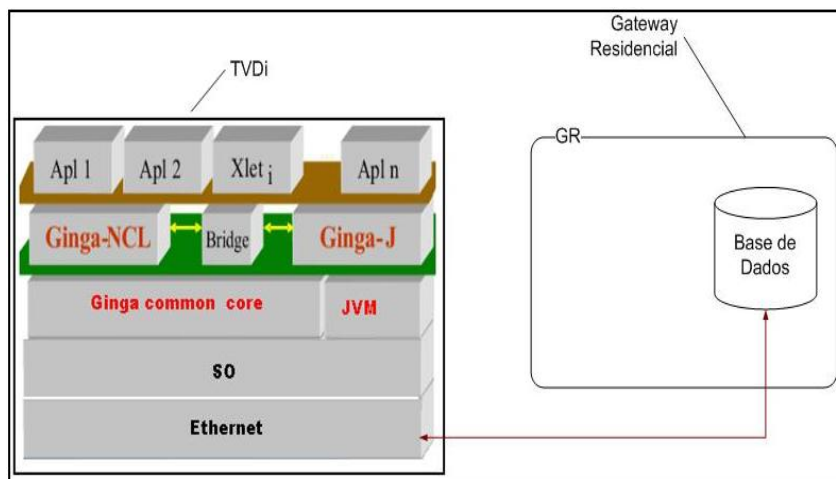


Figura 3.8: Arquitetura do Módulo TVDi

Fonte: Do próprio autor.

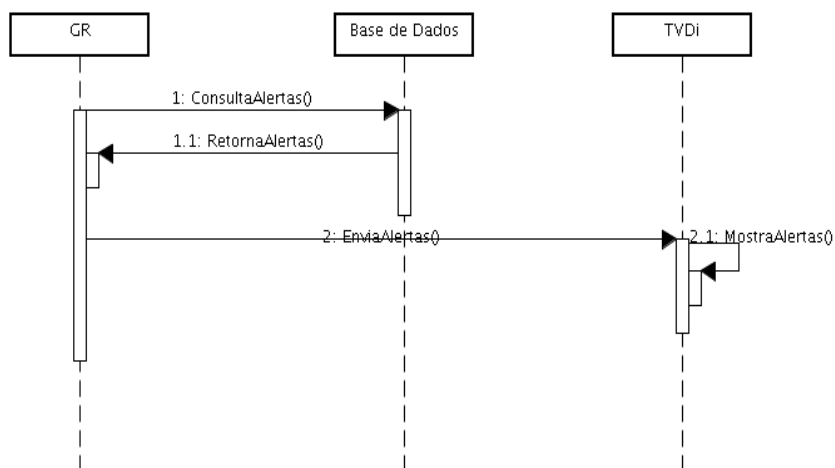


Figura 3.9: Diagrama de Sequencia Envia Alerta para a TVDi

Fonte: Do próprio autor.

Tabela 3.7: Cenário para a Apresentação de Alertas na TVDi

Nome do Cenário	Apresentação de Alertas na TVDi
Resumo	A aplicação da TVDi deve está executando em background e o STB conectado à rede residencial
Fluxo normal de eventos	1. Sistema verifica alertas na base de dados 2. Sistema envia alertas para a TVDi via rede 3. A TVDI apresenta o alerta na tela
Fluxo alternativo de eventos	Nenhum
Exceções no fluxo normal de eventos	No passo 2, caso não exista alertas na base de dados o sistema não envia dados para a TVDi
Estado anterior ao inicio do cenário	Sistema pronto para ler a base de dados
Estado posterior ao cenário	Dados apresentados na tela da TVDi
Estímulo	Sistema incializa automaticamente
Pré-condições	Sistema encontra-se operacional
Pós-condições	Sistema envia dados para a TVDi e a TVDi apresenta os dados na tela
Cenário desenvolvido por	Vandermi Silva

Como visto na Tabela 3.7, existem dois atores que participam ativamente na arquitetura. São eles o STB com um módulo de apresentação de alertas para a TVDi e o GR, que atua enviando os alarmes pela rede a partir da leitura dos arquivos armazenados na base. Nesse modelo, a TVDi é utilizada como interface de visualização dos alarmes e o processamento das mensagens é feito pelo GR, para não sobrecarregar o STB. Caso o usuário necessite interagir com o sistema de alarmes, por exemplo, apertando a tecla vermelha do controle remoto, o módulo responde fechando a aplicação.

3.3 Conclusão

Nesse Capítulo, foram apresentados os conceitos arquiteturais do trabalho e os cenários de uso de cada módulo da arquitetura proposta. Diagramas de sequência foram utilizados para demonstrar a comunicação entre os módulos, bem como figuras

representativas de cada módulo da arquitetura e sua forma de comunicação foram mostradas.

Os cenários apresentados auxiliam na identificação dos requisitos para implementação dos protótipos que serão apresentados no capítulo 4. Como decisão de projeto, conclui-se que um *gateway* residencial baseado na arquitetura X86 deve ser usado para integrar os dispositivos médicos convencionais com a TVDi, visto que o STB ainda não possui poder de processamento necessário para ser usado como concentrador de dados. Além disso, um celular ligado ao *gateway* permitirá a implementação de um sistema de mensagens SMS que servirá de canal de retorno da aplicação.

Capítulo 4

Implementação do Modelo Proposto de Acordo com os Cenários

Neste Capítulo os cenários mostrados no capítulo 1 serão implementados usando a arquitetura apresentada no Capítulo 3. Inicialmente serão listados os equipamentos utilizados nos protótipos e a infra-estrutura usada para coleta e armazenamento dos dados, bem como as tecnologias utilizadas. Em seguida as implementações dos módulos de coleta armazenamento e apresentação dos dados serão demonstrados finalizando com a conclusão do capítulo.

4.1 Dispositivos e Tecnologias Utilizados na Implementação

Na implementação dos protótipos dessa dissertação foram usados componentes e dispositivos de prateleira encontrados no mercado, já homologados pela indústria e vendidos em lojas especializadas. Entre as tecnologias utilizadas pode-se citar a tecnologia XBee usada na comunicação entre o GR e o oxímetro mostrados em Digi (2008) e em Medical (2008), a rede GSM e o arquivo em XML para armazenamento de dados.

O *hardware* do GR utiliza a arquitetura X86 e é composto por um processador Intel Atom Dual-core, uma placa mãe AsRock modelo A945GC equipada com chipset da Intel com placa de rede *gigabit Ethernet* e placas de som e vídeo integrados (INTEL, 2008), 1 GB (gigabyte) de memória RAM e um dispositivo de armazenamento de 8 GB, utilizado para instalar o sistema operacional Linux, aplicativos e sistema de arquivos.

Além da arquitetura X86, foi incorporado no projeto um celular modelo SL75, conectado através da porta serial, para permitir o uso de SMS através da rede de uma operadora de telefonia móvel. O uso do celular como canal de retorno do *gateway*, facilitou a implementação de um sistema de envio de mensagens, pois a API Java para envio de SMS (SMSLIB, 2008), otimizou o tempo de desenvolvimento do sistema de mensagens.

Para o dispositivo de coleta de dados da frequência cardíaca, foi utilizado um oxímetro de pulso descrito em Medical (2008), que é um equipamento utilizado para a medição da taxa de oxigênio e frequência cardíaca, através do uso de luz infravermelha, enquanto que para a medição da pressão arterial, foi utilizado um medidor de pressão automático capaz de enviar os dados para o GR coletados através da porta USB. Na Figura 4.1 observa-se o computador configurado como GR e os sensores utilizados no trabalho.



Figura 4.1: Equipamentos Utilizados

Fonte: Do próprio autor.

4.1.1 Configuração e Instalação do Gateway Residencial (GR)

Na configuração do GR foram usadas as distribuições Linux Debian (MCCARTY, 1999) e Ubuntu (VUGT, 2007), como sistemas operacionais. O motivo para usar o sistema operacional Linux no GR se deu pelo fato de que o mesmo não necessita de pagamento de licenças de uso, diminuindo assim os custos com sistema operacional e consequentemente o custo total da solução.

As primeiras implementações do GR foram feitas com base no Debian que foi compilado e instalado apenas com os pacotes e funcionalidades necessárias para a aplicação que incluiu uma JVM na versão 6.0, a biblioteca Biblioteca escrita em Java para transmissão e recepção de dados via porta serial (RXTXJava) para comunicação com a porta serial do GR, a biblioteca SMSLIB para comunicação com o celular, um servidor Rede de computadores de alcance mundial (WEB) Apache e um interpretador de *scripts* PHP.

Essa distribuição Debian compilada para o GR, permitiu que ele fosse instalado em um *pendrive* de 5 GB que posteriormente foi reduzido para 2 GB, para delimitar as características mínimas de hardware necessárias para o GR.

Após os testes com o Debian, foi feita uma instalação similar usando a distribuição

Linux Ubuntu, no mesmo hardware do GR para testar os protótipos. As únicas diferenças entre a instalação das duas distribuições foram no hardware, pois o sistema operacional Ubuntu foi instalado no disco rígido do GR e não no *pendrive*, e a outra foi a interface gráfica, que na distribuição Debian não foi instalada.

4.1.2 Configuração dos Dispositivos de Coleta de Dados

Os dispositivos médicos convencionais escolhidos para a implementação dos protótipos dessa dissertação foram um oxímetro de pulso da NONIM Medical e um medidor automático de pressão da Powerpack, ambos à venda no mercado local.

Para medição dos dados da frequência cardíaca foi utilizado o oxímetro de pulso, um módulo Sensor que utiliza a tecnologia Zigbee (XBee) modelo CON-USB da Rogercom, que converte a interface serial para USB e usa o protocolo 802.15.4 descrito em (80215-4, 2008), que foi conectado a porta USB do GR. A Figura 4.2 ilustra os equipamentos usados no módulo de medição.

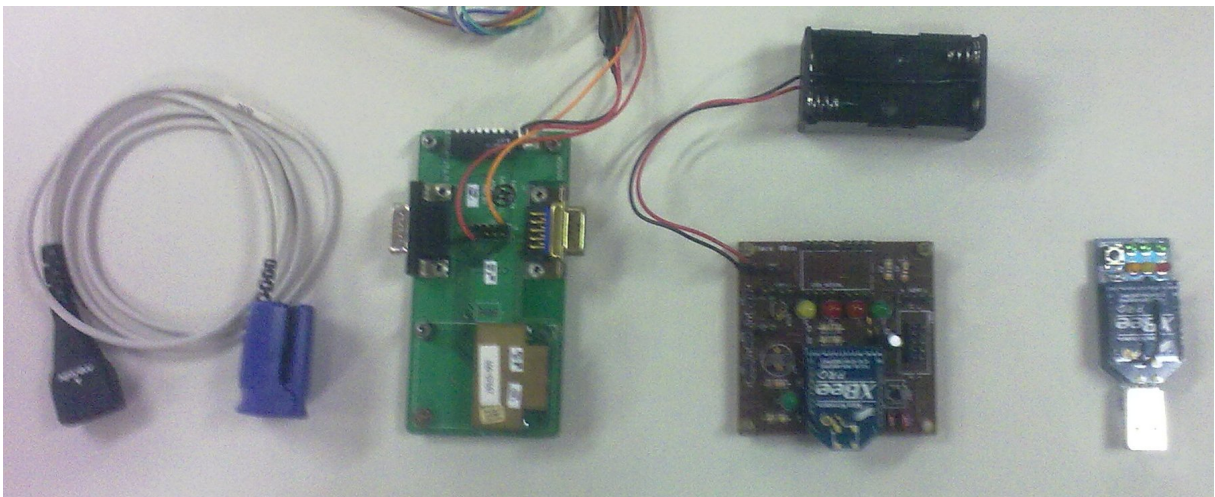


Figura 4.2: Equipamentos Utilizados no Módulo de Coleta

Fonte: Do próprio autor.

Os algoritmos para medição da pressão já estão implementados no sistema embarcado do medidor de pressão, não sendo necessária a implementação neste trabalho. A Figura 4.3, ilustra o equipamento utilizado.



Figura 4.3: Medidor de Pressão Arterial

Fonte: Do próprio autor.

4.2 Implementação dos Cenários

No Cenário 1, um sistema de alerta visual informa ao usuário para medir sua pressão. Nesse caso o sistema usa a TVDi como interface com o usuário e solicita que o mesmo ligue o dispositivo de medição e inicie a medição. Se o usuário não medir a pressão, o alerta permanece na tela até que a mesma seja efetuada. Após a medição o sistema envia uma mensagem para os celulares cadastrados, informando a situação.

No Cenário 2, um oxímetro faz a coleta de dados do usuário e os envia para um GR que se encarrega de interpretar os dados e formatá-los para serem exibidos na TVDi. Algumas situações podem ser observadas nesse cenário para auxiliar na geração de alarmes e construção de regras como por exemplo, as listadas a seguir:

1- Após a medição, se os batimentos cardíacos mantêm-se nos níveis aceitáveis determinados em Camarda e Pinto (2005) o sistema permanece aguardando e comparando as leituras.

2- Se os batimentos ultrapassarem ou ficarem abaixo dos níveis aceitáveis, o sistema aciona uma mensagem de alerta e envia para a TVDi.

Os cenários apresentados podem ser observados na Figura 4.4, que mostra a conexão entre os componentes de rede sem fios, o GR e a TVDi. Usando o celular como canal de retorno da aplicação, o sistema de mensagens do GR recebe mensagens de alerta e as transmite via rede GSM usando mensagens SMS.

No processo de coleta de dados, o oxímetro e o medidor de pressão conectados ao usuário colhem as amostras e as enviam para o GR através de um aplicativo coletor que monitora a porta serial. Um subsistema instalado no GR trata as informações, gerando uma base de dados comum a todos os dispositivos da casa usando um arquivo no formato

XML.

Os dispositivos móveis como, por exemplo, telefones celulares, acessam a base de dados e apresentam as informações na tela do usuário. Este acesso à base a partir de um celular, ocorre quando o usuário se conecta através da rede GPRS da operadora e acessa os sensores disponíveis na residência.

Além do acesso móvel via GPRS, o GR envia uma mensagem no formato SMS para os dispositivos móveis cadastrados e para a TVDi através da rede residencial e do celular conectado a ele.

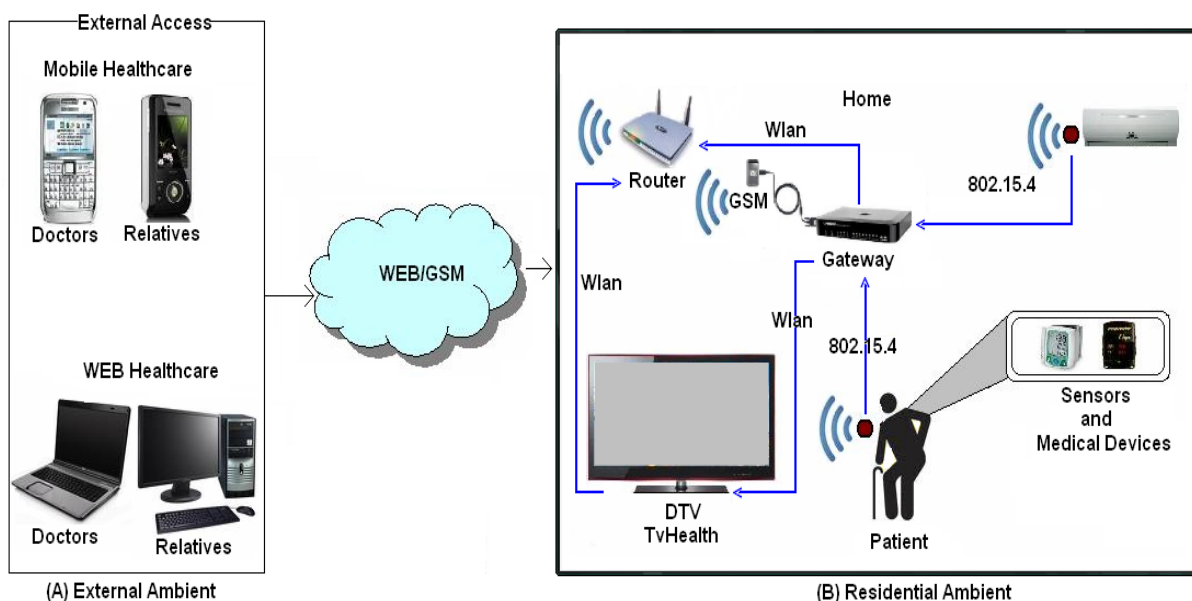


Figura 4.4: Cenário de um Serviço de Coleta e Tratamento de Dados de Sensores.

Fonte: Do próprio autor.

A infraestrutura de redes residenciais apresentada no cenário, é uma mesclagem de redes com e sem fios, na qual é possível estabelecer uma rede de sensores e conectá-los diretamente ao nó sensor do GR. Deste modo, pode-se criar um ambiente residencial monitorado e, ao mesmo tempo, disponibilizar os dados na WEB, usando um roteador de *internet*. Nos próximos tópicos os módulos implementados serão apresentados separadamente.

4.2.1 Implementação do Cenário de Coleta de Dados da Pressão Arterial

Para a implementação do cenário da coleta de dados da pressão arterial, foi necessário usar um dispositivo medidor automático conectado ao GR através de um cabo

USB. Desta maneira, a coleta é feita após o medidor enviar os resultados via porta USB que são tratados pelo GR e disponibilizados para a TVDi e para o celular. O diagrama de caso de uso, classes e trechos do código fonte do protótipo serão apresentados a seguir.

4.2.1.1 Diagrama de Caso de Uso da Coleta de Dados da Pressão Arterial

Nesse cenário o primeiro passo foi desenvolver o diagrama de caso de uso para auxiliar na identificação das classes de coleta. A Figura 4.5 ilustra os atores GR e Pressão e os casos de uso necessários para o desenvolvimento das classes do protótipo. Nota-se que o ator Pressão representa o dispositivo medidor e portanto, apenas envia os dados usando a porta USB enquanto o GR gerencia a coleta e gera os arquivos necessários para a aplicação.

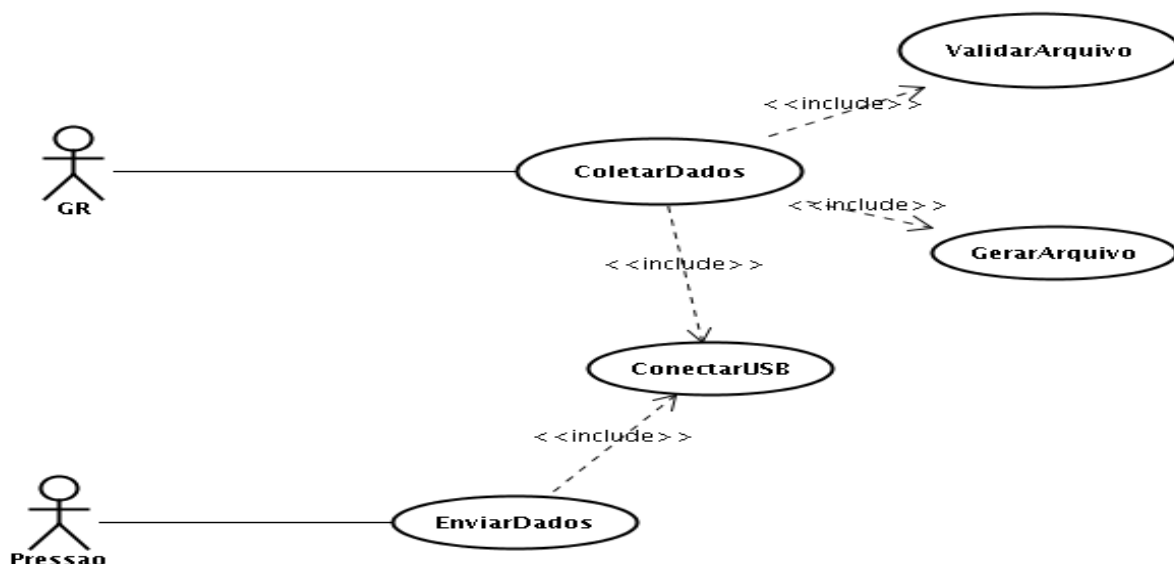


Figura 4.5: Diagrama de Casos de Uso Coleta da Pressão Arterial

Fonte: Do próprio autor.

4.2.1.2 Diagrama de Classe da Coleta de Dados da Pressão Arterial

A partir do diagrama de caso de uso, foi feita a identificação de classes e a divisão dos pacotes para implementação do protótipo que foram identificadas e modeladas no diagrama apresentado na Figura 4.6. As classes fazem parte dos pacotes de *UI*, *dados*, *util* e *dispositivos* e compartilham uma base comum escrita em XML de acordo com o padrão da W3C descrito em McLaughlin (2001).

A classe *ParserDados*, através do método *getXMLPressure*, usa a API JDOM (MCLAUGHLIN, 2001), para gerar o arquivo XML que armazena os dados do medidor

de pressão e o disponibiliza para ser acessado pelos outros módulos.

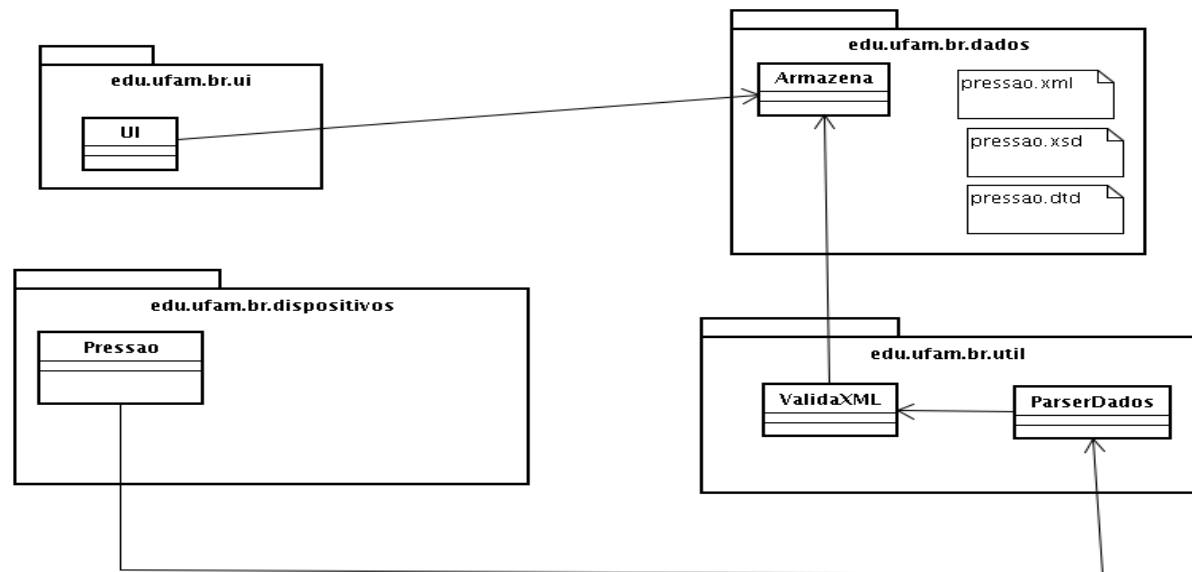


Figura 4.6: Diagrama de Classe Coleta da Pressão Arterial

Fonte: Do próprio autor.

A Figura 4.7 apresenta um trecho do código fonte do método implementado em Java. O método gera o documento XML a partir dos dados recebidos nas variáveis *nome*, *sist* e *dist*, e imprime em arquivo usando a classe do pacote *java.io*.

```

public Element getXMLPressure(String nome, String sist, String dist)
    throws Exception {

    // variavel de instancia de Document XML Jdom

    Element pressure = new Element("pressure");
    Document doc_pressure = new Document(pressure);

    Element pacient = criarElementXml(nome, sist, dist);
    pressure.addContent(pacient);
    // Classe responsável para imprimir / gerar o XML
    XMLOutputter xout = new XMLOutputter();
    try {
        // Criando o arquivo de saída
        FileWriter arquivo_alerta = new FileWriter(new File(FILE_PATH));
        // Imprimindo o XML no arquivo
        xout.output(doc_pressure, arquivo_alerta);
    } catch (IOException e) {
        e.printStackTrace();
    }

    return pacient;
}
  
```

Figura 4.7: Trecho do Código Fonte da Classe ParserDados

Fonte: Do próprio autor.

4.2.1.3 Estrutura de Armazenamento de Dados da Pressão Arterial

A estrutura do arquivo XML armazenado é apresentado na Figura 4.8. A Figura 4.8 (A), representa a base de dados do medidor de pressão, que armazena o nome do paciente, pressão sistólica e diastólica, através das *tags* *name*, *sist* e *dist*. A Figura 4.8 (B), representa o descritor da base feito usando um DTD que identifica os elementos contidos no XML e evita que documentos inválidos sejam gravados.

<pre><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE pressure SYSTEM "pressure.dtd"> <sensor> <pressure> <name>Vandermi</name> <sist>130</sist> <dist>70</dist> </pressure> </sensor></pre> <p style="text-align: center;">(A)</p>	<pre><!ELEMENT sensor (pressure)> <!ELEMENT pressure (name, sist, dist)> <!ELEMENT name (#PCDATA)> <!ELEMENT sist (#PCDATA)> <!ELEMENT dist (#PCDATA)></pre> <p style="text-align: center;">(B)</p>
---	---

Figura 4.8: Documento XML e seu Respectivo DTD

Fonte: Do próprio autor.

4.2.2 Implementação do Cenário de Coleta de Dados da Frequência Cardíaca

O cenário para medição da frequência cardíaca foi desenvolvido a partir da avaliação do diagrama de caso de uso apresentado na Figura 4.9, que auxilia na definição de classes e métodos da aplicação e posteriormente, foram identificadas as classes e pacotes para serem implementados.

4.2.2.1 Diagrama de Caso de Uso da Coleta de Dados da Frequência Cardíaca

Neste caso de uso dois atores interagem com o sistema, são eles o GR e o oxímetro. O ator oxímetro através do caso de uso *EnviarDados* se conecta ao GR e transmite os valores coletados. O ator GR por sua vez, trata os dados usando os casos de uso *GerarArquivo* e *ValidarArquivo*.

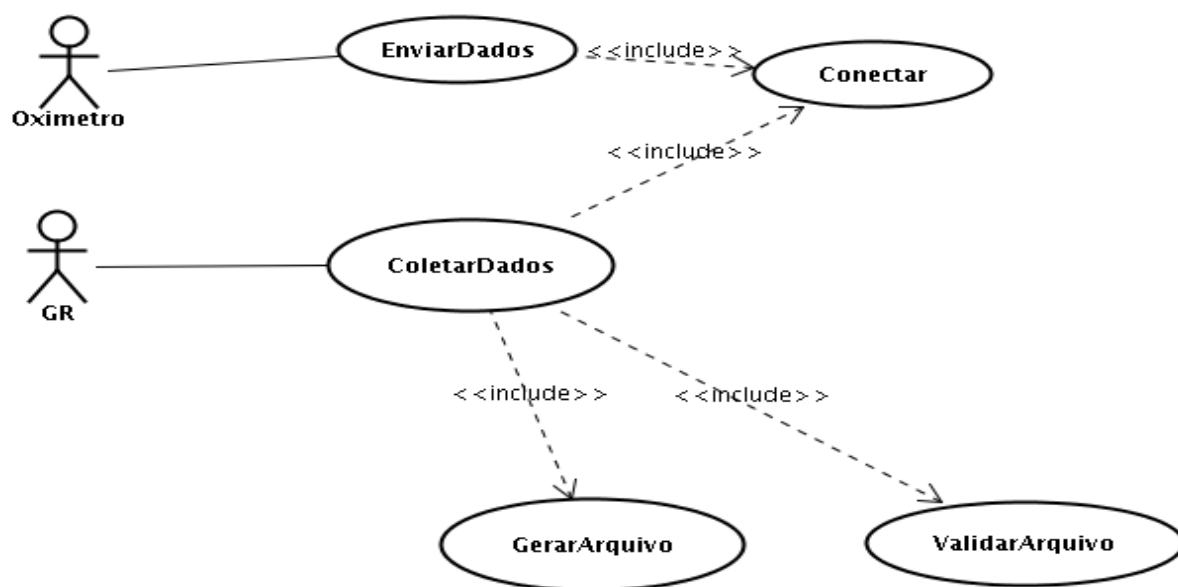


Figura 4.9: Diagrama de Casos de Uso Coleta da Frequência Cardíaca

Fonte: Do próprio autor.

4.2.2.2 Diagrama Classes da Coleta de Dados da Frequência Cardíaca

A partir do caso de uso apresentado na Figura 4.9, foram identificadas as classes para compor o módulo de coleta de dados usando o oxímetro e posteriormente essas classes foram agrupadas em pacotes comuns e implementadas usando a linguagem de programação orientada a objetos Java. O diagrama de classes apresentado na Figura 4.10, mostra as classes e pacotes desenvolvidos nesse módulo. No pacote de *dados* foi desenvolvida uma classe de armazenamento que tem como função manipular os arquivos de dados provenientes do pacote *util*.

O pacote *util* por sua vez, possui as classes *ParserDados* e *ValidaXML*, respectivamente usadas para receber os dados do sensor e validar por meio de um esquema XML ou de um DTD.

No pacote *dispositivos* é implementada a classe *Oxímetro* que é responsável por disponibilizar os dados para a classe *ParserDados* através do método *getDados* responsável por coletar os dados em *bytes* e enviá-los para o *parser* XML.

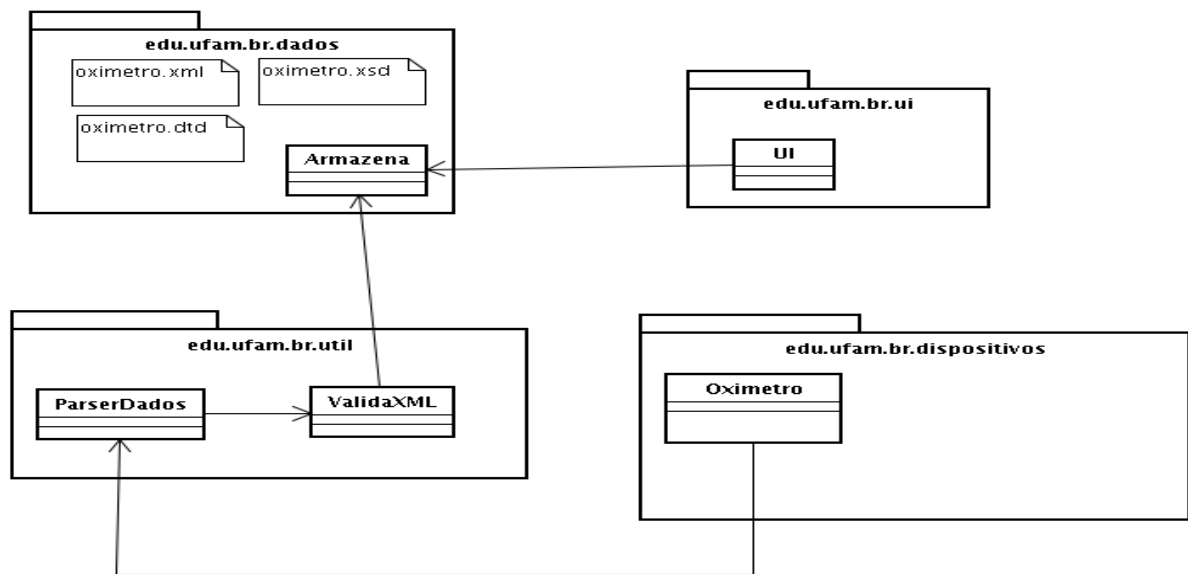


Figura 4.10: Diagrama de Classes e Pacotes Frequência Cardíaca

Fonte: Do próprio autor.

A Figura 4.11 ilustra um trecho do código fonte do método *getDados*, o método coleta os dados em byte e os transforma em uma *string* que é retornada na variável *result*, para posteriormente ser tratada na classe *ParserDados*.

```

//metodo para coletar os dados dos sensores e gerar o xml
public static String getDados(byte[] b) throws Exception {

    String msg = "";
    String result = "";
    String status = "";
    String heart_beat = "";
    String spo2 = "";

    ParserDados.getXMLOximeter(serial_number, msg, heart_beat, spo2);

    // transforma byte em string
    for (int i = 0; i < b.length; i++) {
        result += Integer.parseInt(Integer.toString((b[i] & 0xff) + 0x100,
            16).substring(1), 16);
    }
    .
    .
    .
}
return result;
}
  
```

Figura 4.11: Trecho de Código Fonte do Método GetDados

Fonte: Do próprio autor.

Os dados do oxímetro são lidos em um *array* de bytes e convertidos primeiramente para um conjunto de inteiros e em seguida são divididos em *sub-strings*, para depois serem armazenadas nas variáveis de *status*, *heart_beat*, e *spo2*. Essas variáveis correspondem

respectivamente à situação do sensor, frequência cardíaca e taxa de saturação de oxigênio no sangue (SPO2).

A Figura 4.12 mostra um exemplo dos dados após o tratamento feito pelo método *getDados*, a figura mostra um vetor de inteiros contendo nove posições onde as primeiras três posições correspondem ao *status* do sensor oxímetro por exemplo, se está conectado ao usuário, desconectado ou ativo. As posições três, quatro e cinco armazenam os valores da frequência cardíaca enquanto as posições seis sete e oito armazenam a taxa de SPO2. Os números apresentados na figura, por exemplo, indicam que o oxímetro está desconectado e não está medindo. Esses valores foram identificados a partir do documento do fabricante disponível em Medical (2008).

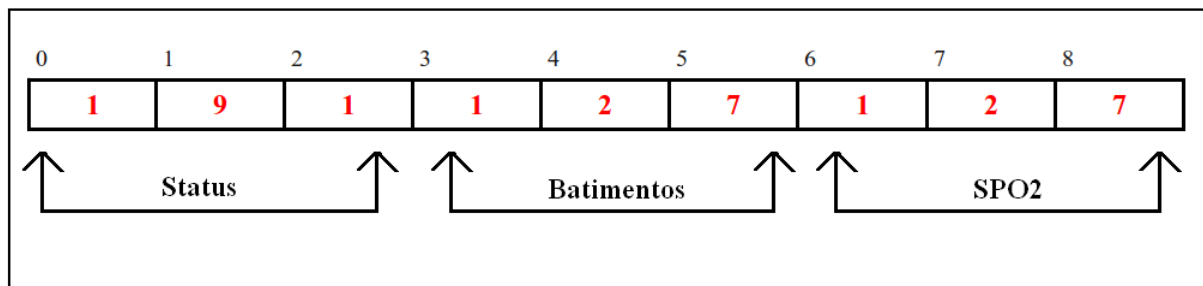


Figura 4.12: Vetor com Dados Coletados do Oxímetro

Fonte: Do próprio autor.

4.2.2.3 Estrutura de Armazenamento de Dados da Frequência Cardíaca

O arquivo XML e seu respectivo DTD são apresentados na Figura 4.13. A Figura 4.13 (A), representa a base de dados do oxímetro e armazena o nome do paciente, estado do sensor, frequência cardíaca e SPO2, através das *tags name, status, heart-beat e spo2*. A Figura 4.13 (B), representa o descritor da base usando um DTD que identifica os elementos contidos no XML.

<pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE oximeter SYSTEM "oximeter.dtd"> <sensor> <oximeter> <name>Vandermi</name> <status>0</status> <heart_beat>85</heart_beat> <spo2>98</spo2> </oximeter> </sensor> </pre> <p style="text-align: center;">(A)</p>	<pre> <!ELEMENT sensor (oximeter)> <!ELEMENT oximeter (name, status,heart_beat,spo2)> <!ELEMENT name (#PCDATA)> <!ELEMENT status(#PCDATA)> <!ELEMENT heart_beat(#PCDATA)> <!ELEMENT spo2(#PCDATA)> </pre> <p style="text-align: center;">(B)</p>
---	---

Figura 4.13: Estrutura do Documento XML da Frequência Cardíaca

Fonte: Do próprio autor.

4.2.3 Implementação do Cenário do Módulo de Envio de Mensagens

O módulo de envio de mensagens SMS, foi implementado para ser usado como canal de retorno dos protótipos de *healthcare* apresentados nessa dissertação e permite que um celular de baixo custo seja utilizado para enviar mensagens sobre a situação dos sensores e dispositivos conectados ao usuário via rede da operadora de telefonia celular. O diagrama de caso de uso, diagrama de classes e partes do código fonte escrito em Java serão apresentados nos tópicos a seguir.

4.2.3.1 Diagrama de Caso de Uso do Módulo de Envio de Mensagens

Este diagrama de caso de uso possui um ator chamado *Mensageiro* e acessa os casos de uso *VerificarAlarme*, *GerarSMS* e *EnviarSMS*. A partir do diagrama de caso de uso mostrado na Figura 4.14, foram identificadas possíveis classes e métodos à serem desenvolvidos no diagrama de classes.

Nesse módulo o celular usado como canal de retorno é acessado via porta USB através de um módulo de mensagens instalado no *gateway* e implementado sobre a biblioteca SMSLIB (SMSLIB, 2008). Desse modo uma classe de envio de mensagens foi desenvolvida para acessar os métodos da biblioteca e enviar as mensagens provenientes do GR.

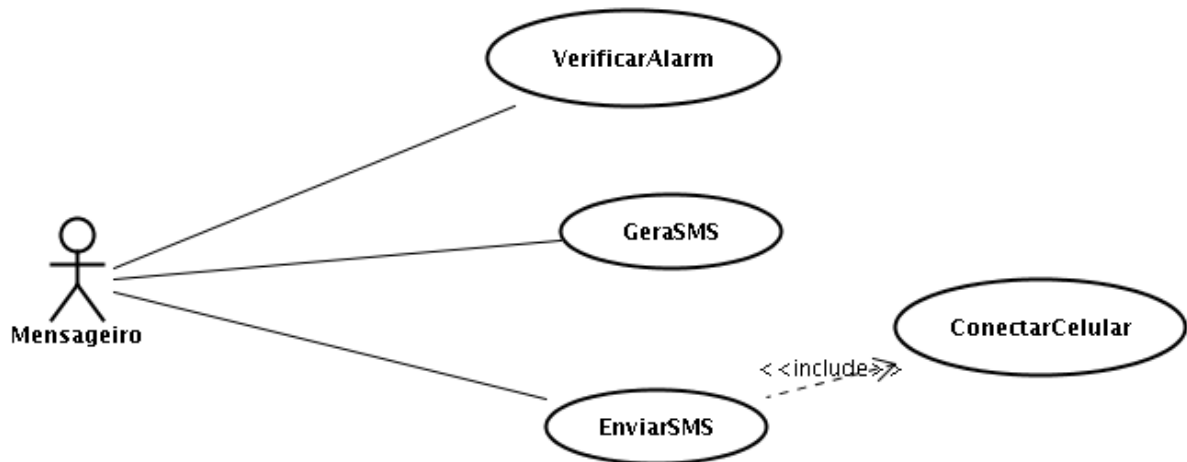


Figura 4.14: Diagrama de Caso de Uso do Módulo de Mensagens

Fonte: Do próprio autor.

4.2.3.2 Diagrama de Classe do Módulo Envio de Mensagens

O diagrama de classes apresentado na Figura 4.15 utiliza o pacote *org.sms.lib* que através da classe *Service* instancia os métodos *setOutboundMessageNotification*, *addGateway* e *startService*, todos necessários para abrir a conexão com o celular, configurá-lo como modem e inicializar os serviços de envio de SMS.

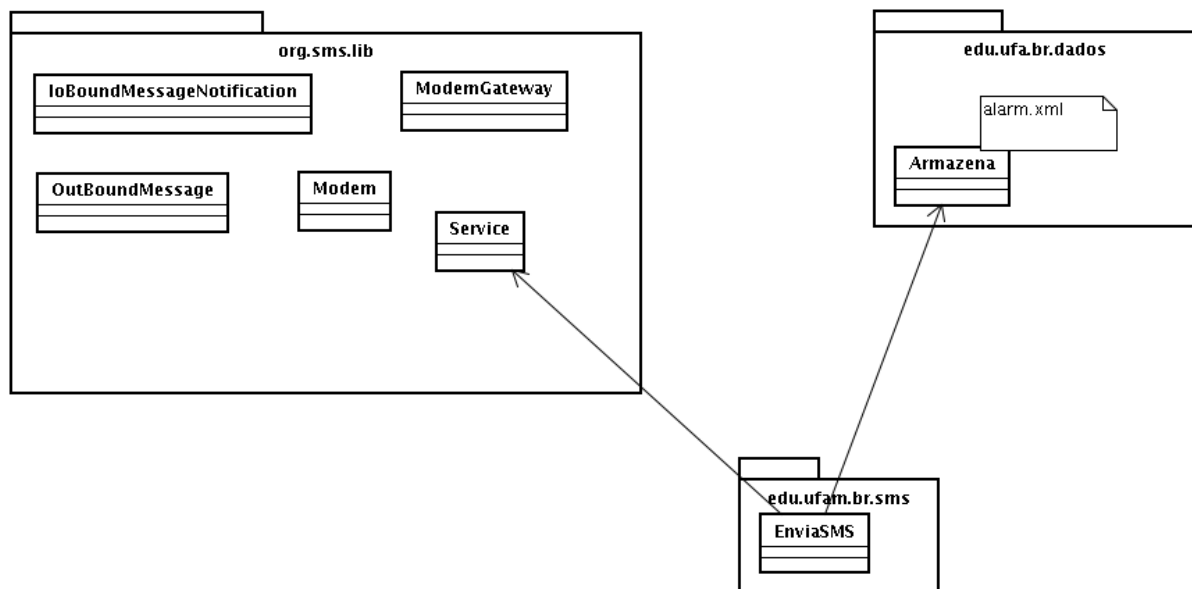


Figura 4.15: Diagrama de Classe do Módulo de Mensagens

Fonte: Do próprio autor.

O trecho do código fonte de envio de SMS é apresentado na Figura 4.16, o serviço de mensagem é encapsulado nos argumentos *nr-celular* e *mensagem*, que recebe o número do celular e o tipo de mensagem, através da leitura do arquivo de alarme em formato XML.

Observando a Figura 4.16 nota-se que a configuração da porta USB utilizada pelo sistema de mensagem para conexão com o celular, é passada como argumento para a classe *SerialModemGateway*, que é a classe responsável por coordenar o acesso às portas seriais.

```
package edu.ufam.br.sms;
import org.smslib.IOutboundMessageNotification;
import org.smslib.OutboundMessage;
import org.smslib.Service;
import org.smslib.modem.SerialModemGateway;

public class EnviaSMS {
    public void doIt() throws Exception {
        Service srv;
        OutboundMessage msg;
        OutboundNotification outboundNotification = new OutboundNotification();
        srv = new Service();
        SerialModemGateway gateway = new SerialModemGateway("modem.com1",
            "/dev/ttyACM0", 9600, "Nokia", "Xpress");
        gateway.setInbound(true);
        gateway.setOutbound(true);
        gateway.setSimPin("0000");
        srv.setOutboundMessageNotification(outboundNotification);
        srv.addGateway(gateway);
        srv.startService();
        // Send a message synchronously.
        msg = new OutboundMessage( nr_celular, mensagem );
        srv.sendMessage(msg);
        srv.stopService();
    }
}
```

Figura 4.16: Trecho de Código do Módulo de Envio de Mensagens

Fonte: Do próprio autor.

Um exemplo de um arquivo de alerta em formato XML é apresentado na Figura 4.17, as tags *pacient*, *doctor-phone* e *message* são extraídas do documento de dados dos usuários para em seguida fazerem parte do documento de alarme.

```
<?xml version="1.0" encoding="UTF-8"?>
<Alerta>
<message_data>
  <pacient>Vandermi</pacient>
  <doctor_phone>81252248</doctor_phone>
  <message>Pacient: Vandermi Batimentos: 120 Oxigeny: 90</message>
</message_data>
</Alerta>
```

Figura 4.17: Exemplo de um Alerta em Formato XML

Fonte: Do próprio autor.

4.2.4 Implementação do Cenário de Apresentação na TVDi e no Celular

O módulo de apresentação dos dados para a TVDi, foi desenvolvido usando dois STBs, sendo um com o *middleware* europeu MHP, com suporte a Java-TV e o outro usando a implementação do *middleware* brasileiro GINGA-NCL, disponível em RCAsoft (2008), que permite a execução de aplicações interativas que podem ser executadas de um *pendrive* diretamente no STB ou via *broadcast* das emissoras de TVDi.

Para a implementação da aplicação Java-TV foi usada a linguagem Java e a biblioteca NanoXML (NANOXML, 2008) para leitura dos dados e alarmes enquanto que para as implementações do GINGA-NCL, foram usadas as linguagens de *script* Lua e NCL. A apresentação dos dados no celular utilizou a arquitetura JME e foi desenvolvida de maneira a permitir o acesso externo aos dados do GR usando a rede da operadora.

4.2.4.1 Apresentação na TVDi usando a Especificação Java TV

Inicialmente foi desenvolvido o diagrama de caso de uso identificando possíveis funcionalidades para o módulo de apresentação dos dados na TVDi. Em seguida, as classes foram identificadas e projetadas finalizando com a implementação do *parser* XML e das telas de navegação do módulo.

O diagrama de caso de uso da Figura 4.18, apresenta em alto nível as principais funcionalidades do módulo. Os atores envolvidos nesse caso de uso são a TVDi o usuário e o GR. O ator TVDi recebe e apresenta dados e alarmes disponibilizados pelo ator GR, que gerencia dados e gera os alarmes quando necessário. O ator usuário controla a aplicação através de um controle remoto que lhe permite usar a interatividade proporcionada pela aplicação.

Observando o diagrama desse caso de uso, nota-se que tanto o GR quanto a TVDi compartilham o caso de uso *Verifica dados*, isso acontece porque a base de dados em XML

é compartilhada entre todos os módulos da arquitetura.

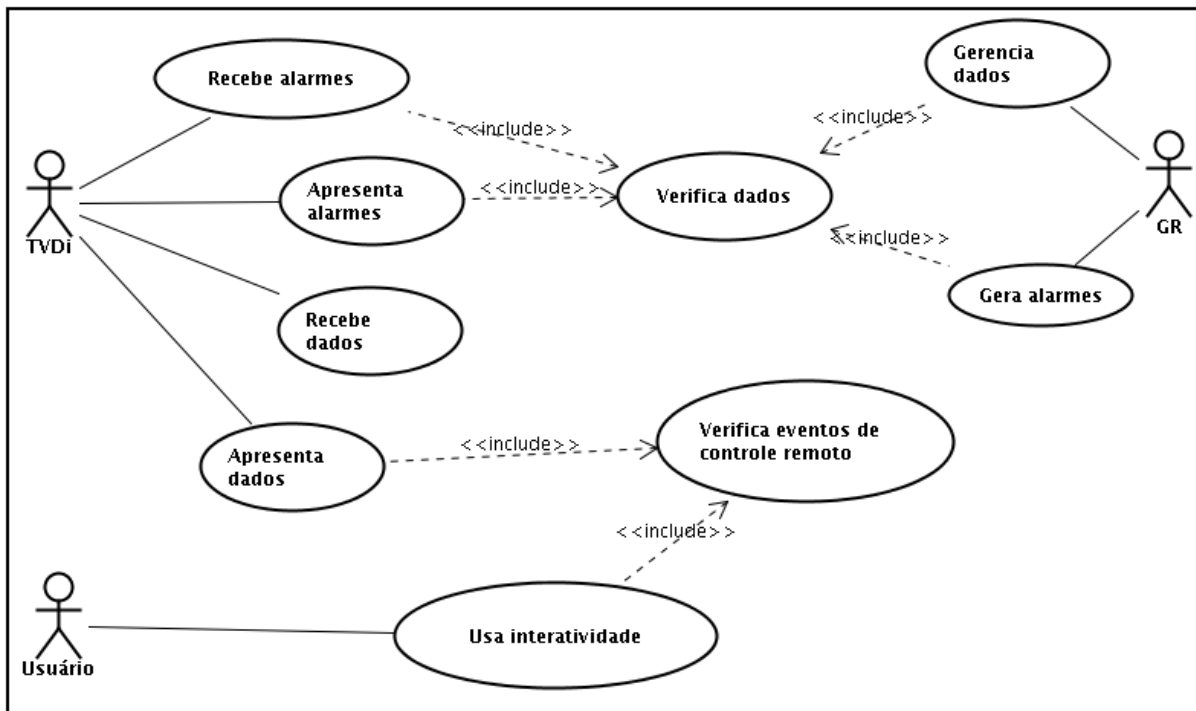


Figura 4.18: Diagrama de Caso de Uso do Módulo TVDi

Fonte: Do próprio autor.

Na Figura 4.19 é possível observar o uso de três pacotes a saber: *xlet*, *screens* e *havi*, o pacote *screen*, contém as classes de telas da aplicação Xlet, sendo uma tela principal, *MainPage*, uma tela de alarme e uma tela de dados, *ScreenData*. No pacote *xlet*, a classe *MonitorSensorsXlet* implementa a interface *xlet* e possui, dentre outros, um método para controle de eventos chamado de *keyPressed*, que através do argumento *KeyEvent key* responde à ação do controle remoto. Essas ações são executadas dentro de uma estrutura de seleção, que recebe o valor da tecla do controle que foi pressionada e executa a ação.

A Figura 4.20 apresenta parte do código fonte desse método implementado e a Figura 4.21 ilustra o diagrama de estados que controla a navegação entre as telas da aplicação.

O diagrama de estados da Figura 4.21 ilustra a navegabilidade entre as classes de interatividade, dados e alarme e mostra a mudança de estado da aplicação quando uma tecla do controle remoto é pressionada. Por exemplo, um estado em que o usuário, a partir da tela de interatividade, pressione a tecla vermelha do controle remoto, a aplicação responde mudando de estado e apresenta a tela de dados com as últimas medições

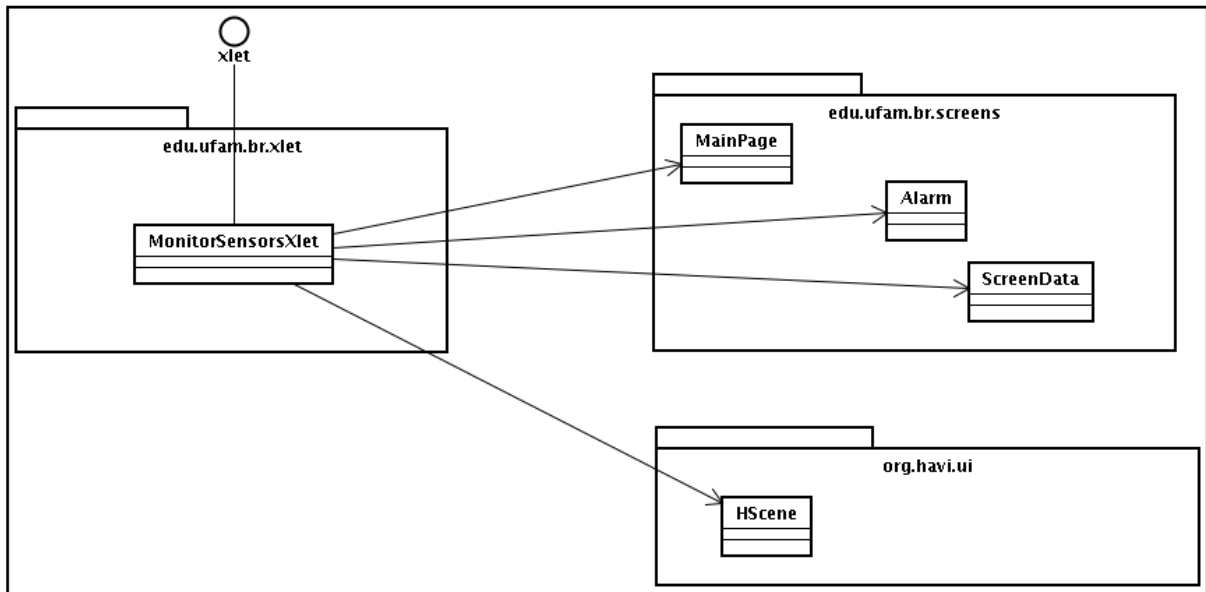


Figura 4.19: Diagrama de Classes do Módulo TVDi

Fonte: Do próprio autor.

```

/*
 * tratamento de eventos dos botões vermelho, verde e azul respectivamente
 * monitoramento, contatos e fechar
 *
 * @see java.awt.event.KeyListener#keyPressed(java.awt.event.KeyEvent)
 */
// Controle dos estados da aplicação
public void keyPressed(KeyEvent key) {

    // pega o codigo da tecla do controle remoto
    int codigo = key.getKeyCode();
    switch (codigo) {
    case 403: // botao vermelho do controle remoto
        telainteratividade.setVisible(false);
        dadosSensores.setVisible(true);
        repaint();
        break;
    case 406:// botao azul do controle remoto
        dadosSensores.setVisible(false);
        telainteratividade.setVisible(true);
        repaint();
        break;

    case 27: // botão exit do controle remoto

        destroyXlet(true);
        break;
    default:

    }
}
}
  
```

Figura 4.20: Código Fonte do Método *keyPressed*

Fonte: Do próprio autor.

coletadas pelo sensor e disponibilizadas pelo GR.

Outro exemplo tratado no diagrama de estados, é o estado de alarme da aplicação. Por exemplo, se o GR enviar um alarme para a TVDi, imediatamente o estado passa de *noAlarm* para *Alarm*, e aciona a classe *ApAlarme* para que as ações sejam tomadas.

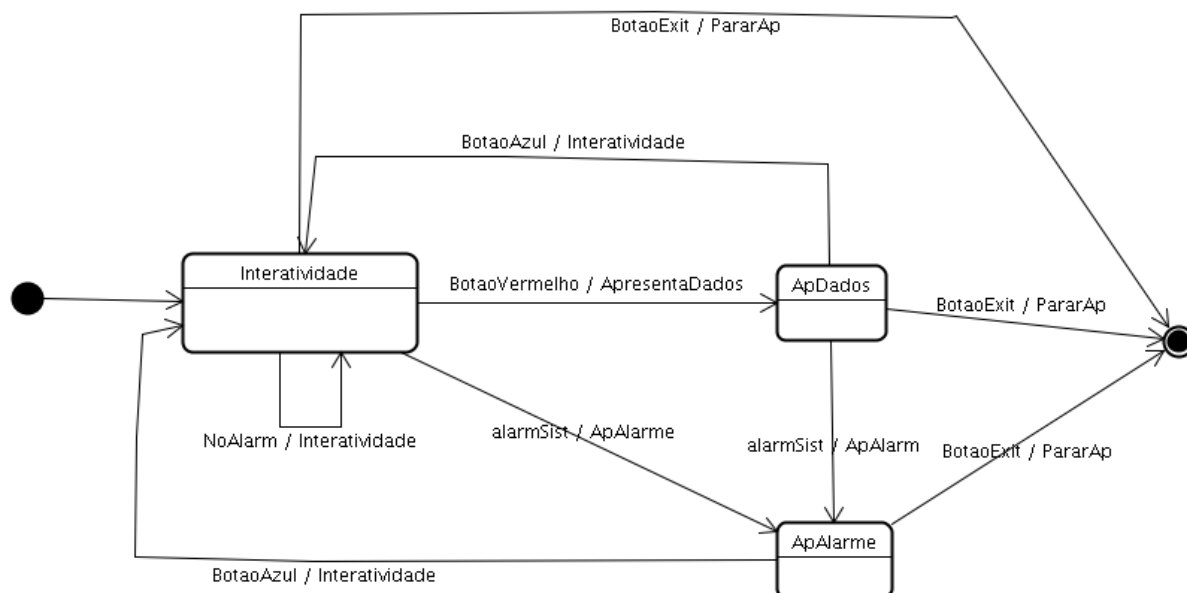


Figura 4.21: Diagrama de Estados da Aplicação

Fonte: Do próprio autor.

Esse módulo foi testado e executado usando o emulador XletView e um STB comercial, que possui a implementação do *middleware* MHP. Os resultados dos testes serão apresentados no Capítulo 5 dessa dissertação.

4.2.4.2 Apresentação na TVDi usando a Especificação Ginga-NCL

Atualmente no Brasil, a maioria dos STB compatíveis com o padrão brasileiro ainda não executam código escrito em Java, porém, algumas empresas como por exemplo a RCAsoft possuem uma implementação do *middleware* do sistema brasileiro que permite por enquanto a execução de aplicações escritas em NCL e em Lua. Essa implementação do GINGA-NCL está embarcada em um STB comercial e permite que aplicações interativas sejam executadas através de um *pendrive* conectado na porta USB.

A fim de testar o comportamento do módulo de apresentação na TVDi, foi desenvolvida uma aplicação em NCL e Lua que usa a arquitetura do GR descrita nessa dissertação. Através da avaliação do caso de uso mostrado na Figura 4.18, o diagrama de classes foi adaptado para Lua e NCL conforme é apresentado na Figura 4.22.

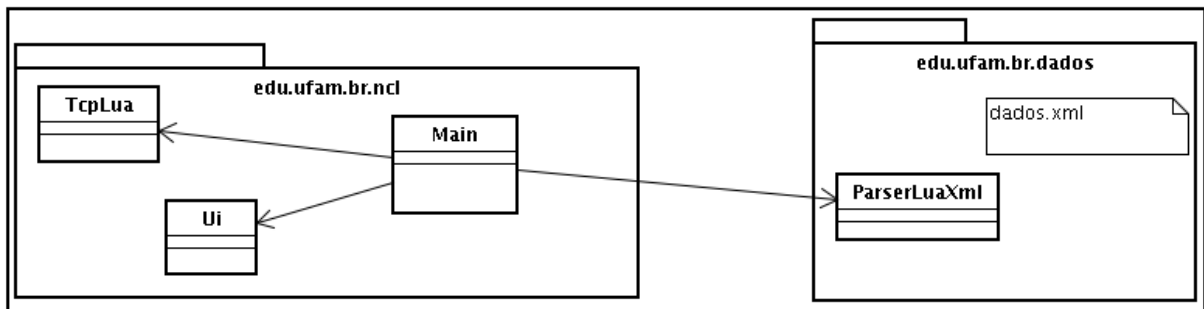


Figura 4.22: Diagrama de Classes Adaptado para Lua e NCL

Fonte: Do próprio autor.

A classe Main, faz chamadas para as classes TCPLua, Ui e ParserLuaXML. A classe TCPLua, é responsável pela conexão com o GR através da rede local, a classe ParserXML é responsável por ler os dados disponibilizados pelo GR e a classe Ui formata e apresenta os dados na tela da TVDi.

```

collect = function(s)
  local stack = {}
  local top = {}
  table.insert(stack, top)
  local ni,c,label,xarg, empty
  local i, j = 1, 1
  while true do
    ni,j,c,label,xarg, empty = string.find(s, "<(/?)(%w+)(.)(/?)>", i)
    if not ni then break end
    local text = string.sub(s, i, ni-1)
    if not string.find(text, "^%s*$") then
      table.insert(top, text)
    end
    if empty == "/" then -- empty element tag
      table.insert(top, {label=label, xarg=parserXML.parseargs(xarg), empty=1})
    elseif c == "" then -- start tag
      top = {label=label, xarg=parserXML.parseargs(xarg)}
      table.insert(stack, top) -- new level
    else -- end tag
      local toclose = table.remove(stack) -- remove top
      top = stack[#stack]
      if #stack < 1 then
        error("nothing to close with "..label)
      end
      if toclose.label ~= label then
        error("trying to close "..toclose.label.." with "..label)
      end
      table.insert(top, toclose)
    end
    i = j+1
  end
  local text = string.sub(s, i)
  if not string.find(text, "^%s*$") then
    table.insert(stack[#stack], text)
  end
  if #stack > 1 then
    error("unclosed "..stack[stack.n].label)
  end
  return stack[1]
end

```

Figura 4.23: Trecho de Código Fonte em LUA

Fonte: Do próprio autor.

A Figura 4.23 mostra a implementação de parte da classe ParserLuaXml onde a

classe usa uma função para carregar o arquivo XML e passa como argumento um conjunto de *Strings*, que são inseridas em uma tabela padrão Lua, que usa um algoritmo de pilha para ler os dados do arquivo XML e disponibilizá-los para a classe Ui.

4.2.4.3 Apresentação no Celular Usando JME

O módulo de visualização dos dados via celular, foi baseado no diagrama de caso de uso e diagrama de classes mostrados nas Figuras 4.24 e 4.25.

O diagrama de caso de uso mostra a interação entre o usuário, o celular e o GR, e foi utilizado para definir o diagrama de classes da aplicação móvel.

A aplicação móvel funciona da seguinte maneira:

- o usuário executa a aplicação móvel
- a aplicação abre uma conexão *Hypertext Transfer Protocol* (HTTP) com o GR através da rede GPRS
- a aplicação lê os dados em XML e faz o parser no próprio dispositivo móvel.

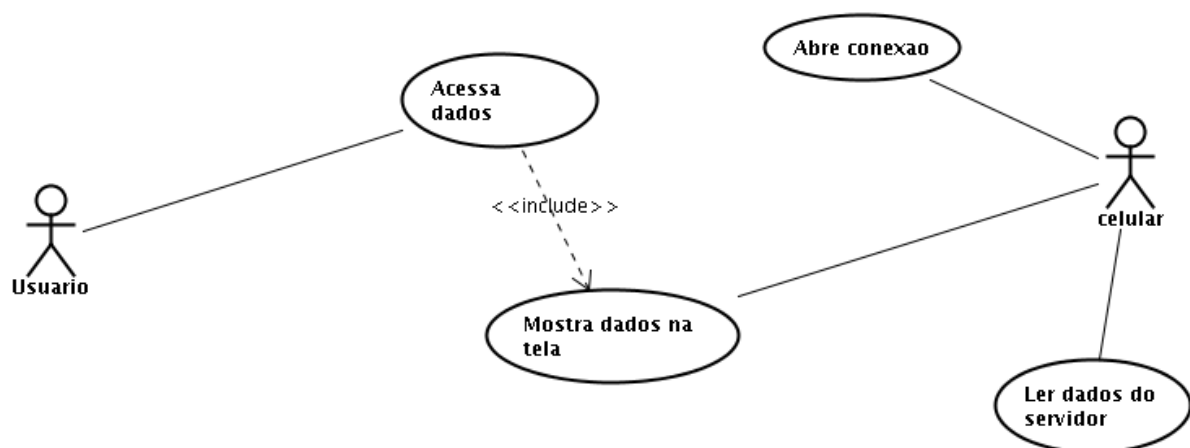


Figura 4.24: Diagrama de Casos de Uso para a Aplicação Móvel

Fonte: Do próprio autor.

O diagrama de classes apresentado na Figura 4.25 mostra a interconexão entre os pacotes de dados e o *mobile*. A classe *M.health*, usar o perfil CLDC e as classes *XMLGetOximeter* e *XMLGetPressure*, que fazem o *parser* do arquivo XML disponibilizado pelo GR.

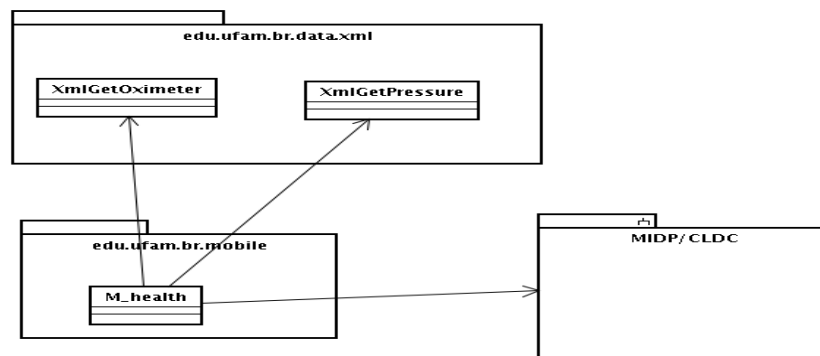


Figura 4.25: Diagrama de Classes para a Aplicação Móvel

Fonte: Do próprio autor.

A Figura 4.26 mostra uma parte do código fonte implementado em JME que executa embarcado no celular. No código apresentado, é usada a biblioteca KXML (KXML, 2008) utilizada para fazer o *parser* do arquivo XML disponibilizado na base de dados do GR.

```

/**
 * @author Vandermi Silva
 * Classe usada para realizar o parser no arquivo XML
 */
public class XMLGetPressure {

    public String parser(InputStream in) throws Exception {

        String temp = "";
        KXmlParser parser = new KXmlParser();
        parser.setInput(new InputStreamReader(in));
        parser.nextTag();
        //Posiciona na tag <sensors>
        parser.require(XmlPullParser.START_TAG, null, "pressure"); //sensors
        //Enquanto é diferente de END_TAG
        while (parser.nextTag() != XmlPullParser.END_TAG) {
            //Posiciona na tag <sensor>
            parser.require(XmlPullParser.START_TAG, null, "data");//sensor_data
            temp = parserTemp(parser);
            parser.require(XmlPullParser.END_TAG, null, "data"); //sensor_data
        }

        parser.require(XmlPullParser.END_TAG, null, "pressure");//sensors
        parser.next();
        parser.require(XmlPullParser.END_DOCUMENT, null, null);
        return temp;
    }
}
  
```

Figura 4.26: Parte do Código Fonte da Classe XMLGetPressure

Fonte: Do próprio autor.

4.3 Conclusão

Nesse capítulo foram apresentadas as implementações de protótipos que executam no GR, na TVDi e no celular usando a arquitetura proposta no trabalho. Foram usados

diagramas de caso de uso, classes e um diagrama de estados para demonstrar a navegação entre as telas do protótipo executado na TVDi. Também foram apresentadas e explicadas as partes principais do código fonte dos protótipos.

O objetivo principal desse capítulo foi apresentar a integração entre a TVDi e um *gateway* residencial para coleta de dados de dispositivos médicos convencionais, usando a infra-estrutura de rede existente na residência do usuário e disponibilizar esses dados para acesso via telefone celular.

O protótipo para envio de mensagens SMS, usando um celular de baixo custo, serviu para indicar o seu uso como canal de retorno das aplicações instaladas no *gateway*, desde que seja feito um contrato com operadoras de telefonia móvel afim de garantir o envio e recebimento de mensagens SMS com o mínimo de atraso.

O protótipo de apresentação na TVDi e no celular, permitiu lidar com restrições de *hardware*, pois os dispositivos utilizados ainda não possuem memória e processamento suficiente para executar aplicações mais robustas. Desta forma, o uso de um *gateway* para coletar e tratar os dados torna-se interessante para integrar esses dispositivos e permitir o acesso à visualização dos resultados na tela da TV e do celular.

No Capítulo 5 serão apresentados os testes e resultados dos protótipos executando em emulador e no STB real.

Capítulo 5

Testes e Resultados

Esse capítulo tem como propósito apresentar os resultados e testes efetuados com os protótipos em um STB real e em um ambiente emulado. No decorrer do capítulo, serão apresentados os equipamentos utilizados nos testes e considerações sobre cada módulo implementado.

5.1 Cenário Medir Pressão

O cenário no qual um usuário mede a pressão arterial, apresentado no Capítulo 1 é descrito nesse capítulo através de dois protótipos, sendo um implementado no emulador XleTView e outro em um STB comercial.

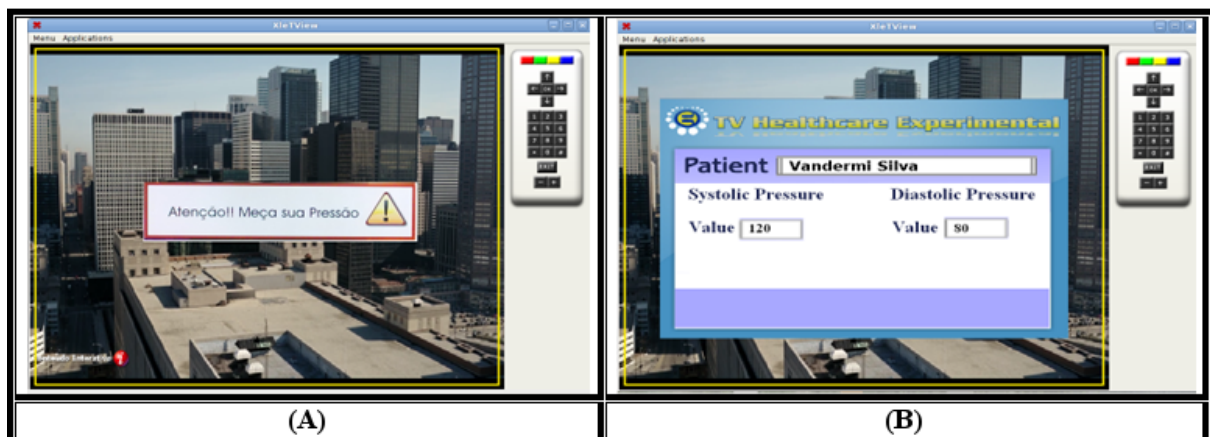


Figura 5.1: Tela de Apresentação de Alarme

Fonte: Do próprio autor.

Para execução dos testes foi necessário alimentar o sistema com dados de usuários e regras de medição da pressão arterial e frequência cardíaca. Em seguida o oxímetro e o medidor de pressão foram conectados ao usuário e o sistema foi inicializado.

Neste cenário o usuário recebe um alerta através da TVDi informando que ele deve medir sua pressão arterial, e a mensagem permanece na tela até que o usuário ligue o medidor. A Figura 5.1 (A) ilustra o momento em que o aviso é apresentado na TVDi usando um STB real com NCL-LUA. Em seguida, após a medição da pressão o usuário visualiza na tela da TVDi os dados coletados. A Figura 5.1 (B), mostra a apresentação dos dados executando no STB real. O mesmo cenário é apresentado na Figura 5.2 (A) e 5.2 (B), onde o ambiente é emulado através do XleTView.



Figura 5.2: Tela de Apresentação de Dados

Fonte: Do próprio autor.

5.1.1 Dispositivos de Hardware e Software Utilizados

Os dispositivos de *hardware* e *software* utilizado nos testes desse cenário foram:

- Um *Set-Top Box*.
- Um medidor de pressão arterial automático com cabo USB.
- Uma televisão LCD.
- Dois celulares.
- Um computador *Desktop* configurado como GR.
- Emulador XleTView, instalado em um *Desktop*

5.1.2 Considerações sobre o protótipo

O protótipo executando a partir do STB acessou o GR via rede cabeada sem maiores problemas e apresentou os dados do medidor de pressão, contudo, as operações

de *refresh* na tela da TV eram perceptíveis quando se utilizava imagens de fundo com alta resolução. A solução para evitar esse problema foi diminuir a resolução das imagens utilizadas como fundo da aplicação.

A principal dificuldade encontrada foi na coleta de dados feita no *gateway*, pois o fabricante do medidor de pressão não disponibiliza o acesso ao protocolo de comunicação que conecta o medidor com o computador através da porta USB, então foi necessário entender como a cadeia de bytes era enviada para a porta serial e em seguida transformá-la em um documento visível para outros dispositivos.

5.2 Cenário Medir Frequência Cardíaca

Para os testes com o protótipo medidor de frequência cardíaca foi necessário primeiramente configurar os módulos XBee local e remoto de acordo com os parâmetros listados na Figura 5.3. Essas configurações foram necessárias para estabelecer uma conexão ponto a ponto com o dispositivo remoto para receber os dados sem a necessidade de cabo serial.

A configuração da Figura 5.3 (A), corresponde ao módulo Xbee que é conectado ao *gateway* enquanto que a configuração da Figura 5.3 (B), corresponde ao dispositivo remoto, nesse caso o oxímetro.

Descrição	Parâmetros	Descrição	Parâmetros
DESTINATION ADDRESS HIGH (DL)	5678	DESTINATION ADDRESS HIGH (DL)	1234
DESTINATION ADDRESS LOW (MY)	1234	DESTINATION ADDRESS LOW (MY)	5678
COORDINATOR ENABLE (CE)	1-COORDINATOR	COORDINATOR ENABLE (CE)	0-END DEVICE
NODE IDENTIFIER (NI)	GR COLLECTOR	NODE IDENTIFIER (NI)	OXIMETRO
INTERFACE DATA RATE (BD)	3-9006	INTERFACE DATA RATE (BD)	3-9006
API ENABLE (AP)	0-API DISABLE	API ENABLE (AP)	0-API DISABLE
DI00 CONFIGURATION (D0)	2-ADC	DI00 CONFIGURATION (D0)	2-ADC
DI01 CONFIGURATION (DI)	2-ADC	DI01 CONFIGURATION (DI)	2-ADC

(A)
(B)

Figura 5.3: Configuração dos Módulos XBee

Fonte: Do próprio autor.

Após a configuração dos dispositivos Xbee e o coletor inicializado, o oxímetro transmite os dados via Xbee até o GR e o GR disponibiliza os dados para a TV. Caso o usuário aperte o botão vermelho do controle remoto, os dados coletados serão mostrados na TVDi, conforme apresentado na Figura 5.4.



Figura 5.4: Tela de Apresentação de Dados no STB

Fonte: Do próprio autor.

Na Figura 5.5 este mesmo cenário é apresentado executando no ambiente emulado com pequenas diferenças na implementação da interface gráfica.

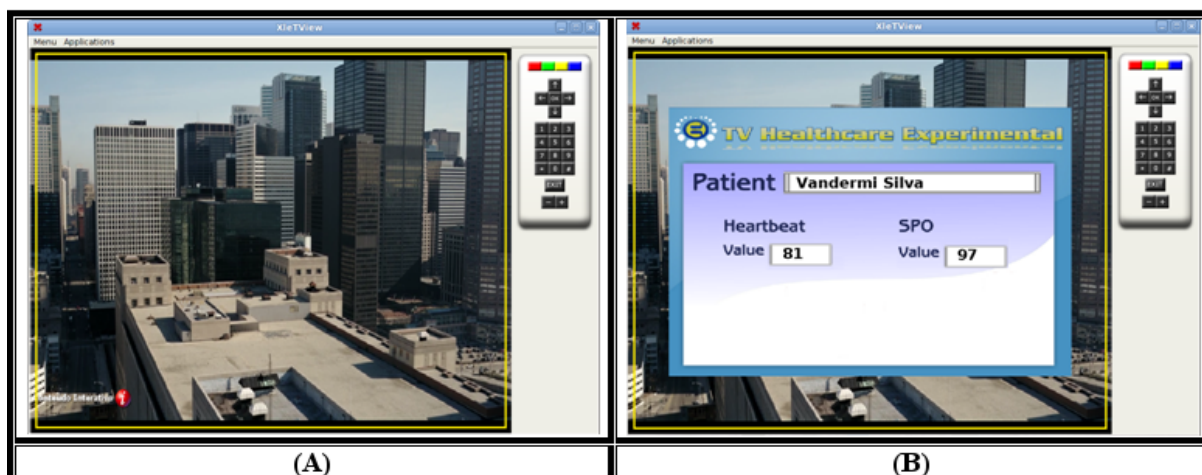


Figura 5.5: Tela de Apresentação de Dados no XletView

Fonte: Do próprio autor.

5.2.1 Dispositivos de Hardware e Software Utilizados

Os dispositivos de hardware e software utilizado nos testes desse cenário foram:

- Um *Set-Top Box*.
- Um oxímetro de pulso.

- Um módulo USB local XBee-Pro série 1 fabricado pela *MaxStream Technology*
- Um módulo remoto XBee-Pro série 1, adaptado em uma placa de circuito
- Uma televisão LCD.
- Dois celulares.
- Um computador *Desktop* configurado como *gateway*.
- Emulador XleTView, instalado em um computador *Desktop*

5.2.2 Considerações sobre o protótipo

Os problemas de *refresh* de tela ocorridos no protótipo de medição da pressão, durante o uso do NCL-LUA e o STB, também ocorreram no protótipo do medidor de frequência cardíaca e o tratamento dado a esse problema foi o mesmo do protótipo do medidor de pressão ou seja, reduzir a resolução das imagens de fundo.

A maior dificuldade encontrada, durante o projeto do protótipo foi na sincronização do oxímetro com o GR, para coletar os dados sem perda significativa. Contudo, após verificação da especificação do fabricante essa questão foi resolvida usando *threads* no protótipo.

5.3 Conclusão

Nesse capítulo foram apresentados os protótipos executando no STB e no emulador XleTView implementados no Capítulo 4.

O objetivo principal desse capítulo, foi mostrar que é possível implementar serviços de *healthcare* para serem acessados pela TVDi, através do uso de um *gateway* residencial. Tanto a aplicação testada no emulador, quanto à testada no STB real, acessaram os equipamentos médicos através da rede residencial satisfatoriamente apesar de serem desenvolvidos em ambientes e linguagens de programação diferentes.

As principais dificuldades encontradas, foram as restrições de hardware tais como, o poder de processamento e a memória por parte do STB e, a sincronização, por parte dos dispositivos medidores de pressão e oxímetro. Contudo essas dificuldades foram superadas, reduzindo o tamanho dos protótipos e suas telas de apresentação.

A plataforma do *gateway* residencial, apresentado nesse trabalho, pode perfeitamente ser usada em trabalhos futuros para melhorar a capacidade de processamento do STB, basta que os dispositivos de hardware necessários, como por

exemplo, receptores de sinal de TVDi, sejam embutidos nele. Desta maneira, o STB poderá gerenciar um sistema de *healthcare*, sem a necessidade de outro dispositivo intermediário.

Para finalizar essa dissertação, no Capítulo 6 serão feitas as considerações finais sobre o trabalho e os pontos de melhorias e sugestões para trabalhos futuros serão apresentados.

Capítulo 6

Considerações Finais

O objetivo geral desse trabalho foi investigar e propor um sistema integrado de informação com características específicas para execução de aplicações de *healthcare*, compatível com o modelo brasileiro de TVDi.

Primeiramente, foram apresentados o problema a motivação a metodologia utilizada e os trabalhos relacionados ao tema, juntamente com os fundamentos de TVDi, *gateway* residencial e dispositivos médicos convencionais. Em seguida, para possibilitar o desenvolvimento da arquitetura necessária à implementação dos serviços de *healthcare*, dois cenários foram criados. Um cenário no qual um medidor de pressão arterial foi usado para medir os valores de pressão de um usuário, e o outro com um oxímetro de pulso para medir a frequência cardíaca do usuário em atividade física.

Durante a fase de concepção da solução, cenários de uso foram utilizados para auxiliar no desenvolvimento dos módulos arquiteturais do sistema e por fim foi descrita a configuração básica para um dispositivo ser usado como *gateway* baseado nos cenários. Em seguida os módulos da arquitetura foram desenvolvidos e implementados usando a infra-estrutura de redes locais dispositivos médicos convencionais e dispositivos Xbee para integrar o gateway com o oxímetro de pulso. E ao final da dissertação, os protótipos de dois serviços de *healthcare* usando a arquitetura proposta, foram apresentados.

6.1 Dificuldades Encontradas

As maiores dificuldades encontradas no decorrer do trabalho e a solução proposta para resolução são listadas a seguir. Entretanto, o trabalho pode ser melhorado e estendido em trabalhos futuros, como por exemplo, testar os protótipos desenvolvidos em Java em um STB real com o GINGA-J implementado, e verificar a possibilidade de embarcar a solução em um STB com maior poder de processamento e memória.

- Acesso aos dados do oxímetro via porta serial - o fabricante não disponibiliza códigos

fontes ou exemplos de implementação tendo em vista que o dispositivo é vendido para consumidores finais. A solução foi desenvolver um leitor de porta serial para identificar a cadeia de bits e fazer o mapeamento para um arquivo XML.

- Acesso aos dados do medidor de pressão via porta serial- o equipamento também é de prateleira e não possui código fonte ou manuais técnicos. A solução foi a mesma usada no oxímetro.
- Sincronização entre os dispositivos e o *gateway* - através de threads escritas em Java, foi possível determinar o tempo que o microcontrolador do oxímetro e do medidor de pressão levavam para disponibilizar os dados na porta serial e assim determinar a leitura da porta serial de um em um segundo feita pelo protótipo de coleta.
- Inexistência de implementação em STB real do sub-sistema GINGA-J para testes dos protótipos - foi necessário o uso de um emulador para Java-Tv e de um STB com o padrão europeu MHP para executar testes de conexão com o *gateway*.
- Integração entre o oxímetro e o dispositivo XBee, pois foi necessário uma placa de circuitos intermediária, desenvolvida no laboratório de eletrônica do curso de Engenharia Elétrica da UFAM e usada neste trabalho para conectar o oxímetro com o *gateway*.

A conclusão final desta dissertação, responde as questões apresentadas nos objetivos específico e mostra uma solução para integrar dispositivos médicos convencionais com a TVDi brasileira, que através do desenvolvimento de uma arquitetura de software para um dispositivo coletor e concentrador de dados intermedia a comunicação entre TV e dispositivos.

6.2 Sugestões para Trabalhos Futuros

Como sugestão para trabalhos futuros, pretende-se estender esse trabalho em pelo menos três pontos a serem melhorados. São eles:

- Desenvolvimento de um sistema que integre a arquitetura com um framework de agentes inteligentes para implementar por meio de inteligência artificial um cuidador virtual de idosos.
- Desenvolver um módulo de coleta totalmente sem fios para permitir maior mobilidade para o usuário.

-
- E estudar uma maneira de embarcar o gateway de healthcare em STB ou em TV com STB integrado

Referências Bibliográficas

80215-4 iee. *IEEE 802.15 WPAN*. Setembro 2008. Acessado em Novembro de 2008. Disponível em: <<http://www.ieee802.org/15/pub/TG4.html>>.

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*. [S.l.], 2007.

ALMEIDA, L.; GARCÍA, L. Aspectos comunicativos, sociais e psicológicos da televisão tradicional e implicações no design de aplicações para o sistema brasileiro de televisão digital. *Workshop em Usabilidade de Aplicações e Tecnologias Emergentes a Necessidade de uma Nova Usabilidade*, p. 11–16, 2006.

AUGUSTO, R. C.; CARLOS, J.; DANIEL, S. Ambient intelligence—the next step for artificial intelligence. *IEEE Intelligent Systems*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 23, n. 2, p. 15–18, 2008. ISSN 1541-1672.

BECKER, V.; SOARES, L. F. G. *Viva Mais Alimentação Saudável*. 2008a. Acessado em Maio de 2009. Disponível em: <<http://clube.ncl.org.br/node/29>>.

BECKER, V.; SOARES, L. F. G. *Viva Mais peso ideal*. 2008b. Acessado em Maio de 2009. Disponível em: <<http://clube.ncl.org.br/node/15>>.

BRANCO, F. de C.; VIANNA, J. M.; LIMA, J. R. P. de. Frequência cardíaca na prescrição de treinamento de corredores de fundo. *R. bras. Ci. e Mov.*, v. 12, n. 2, 2004.

CAMARDA, S. R. de A.; PINTO, V. L. M. Comparação da frequência cardíaca máxima medida com as fórmulas de predição propostas por karvonen e tanaka. *Arq. Bras. Cardiol.*, v. 85, n. 1, 2005.

CHOBANIAN, A. V. et al. Seventh report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure. *the American Heart Association*, v. 1, n. 1, p. 1221–1222, December 2003.

CNDH. *Coordenação Nacional de Hipertensão e Diabetes*. Outubro 2009. Acessado em Dezembro de 2009. Disponível em: <<http://dab.saude.gov.br/cnhd>>.

CORCHADO, J.; BAJO, J.; ABRAHAM, A. Dynamic software product lines. *EEE Intelligent Systems*, v. 23, n. 2, p. 19–25, 2008.

DIGI. *Digi Solutions network-enable medical devices to improve patient care*. Outubro 2008. Acessado em Dezembro de 2008. Disponível em: <<http://www.digi.com/learningcenter/applications/medical>>.

- DUCATEL, K. et al. *Scenarios for Ambient Intelligence (ISTAG Report)*. Seville, ES, 2001.
- FORUM-TVDI. *Forum Brasileiro de TV Digital*. Outubro 2009. Acessado em Dezembro de 2009. Disponível em: <<http://www.forumsbtvd.org.br/>>.
- HARKEY, D.; APPAJODU, S.; LARKIN, M. *Wireless Java Programing for Enterprise Applications Mobiles Devices Go Corporate*. USA: Wiley, 2002. ISBN 0-471-21878-2.
- IBGE. *Pesquisa Nacional por Amostras de Domicílios*. Outubro 2009. Acessado em Novembro de 2009. Disponível em: <<http://www.ibge.gov.br>>.
- IERUSALIMSCHY, R. *Programming in Lua*. Janeiro 2006. Acessado em Maio de 2009. Disponível em: <<http://lua.Org>>.
- INTEL. *Intel atom processors for embedded computing*. Agosto 2008. Acessado em Outubro de 2008. Disponível em: <<http://www.intel.com/design/intarch/atom/index.htm>>.
- JAVA-TV. *Arquitetura da API Java-TV*. Setembro 2008. Acessado em Agosto de 2009. Disponível em: <<http://java.sun.com/javame/technology/javatv/index.jsp>>.
- KXML. *KXML documentation*. Janeiro 2008. Acessado em Maio de 2009. Disponível em: <<http://kxml.sourceforge.net/kxml2>>.
- LIN, W.-W.; YU-HSIANG-SHENG. Using osgi upnp and zigbee to provide a wireless ubiquitous home healthcare environment. *IEEE Computer Society*, p. 268–272, 2008.
- MAIA, O. B. *Uma Infraestrutura de Comunicação entre Dispositivos Domésticos e o Modelo Brasileiro de TV Digital*. Dissertação (Mestrado) — Universidade do Amazonas, 2009.
- MCCARTY, B. *Learning Debian Gnu/Linux*. USA: O’Reilly, 1999. Hardcover. ISBN 978-1-56592-705-6.
- MCLAUGHLIN, B. *Java & XML*. USA: O’Reilly, 2001.
- MEDICAL, N. *Oem-iii-module pulse oximeter*. October 2008. Acessado em Outubro de 2008. Disponível em: <<http://www.nonin.com/OEMSolutions/OEM-III-Module>>.
- MENDES, A. *Arquitetura de software: desenvolvimento orientado para a arquitetura*. Rio de Janeiro,RJ, Brasil: Campus, 2002. Hardcover. ISBN 85-352-1013-X.
- MORRIS, A. S.-C. S. *Interactive TV Standards A guide to MHP, OCAP and Java TV*. Burlington,MA, USA: Elsevier, 2005. 20,60 p.
- NANOXML. *JME documentation*. Janeiro 2008. Acessado em Maio de 2009. Disponível em: <<http://devkix.com/nanoxml.php>>.
- PEKHTERYEV, S. D. G.; H., Z. S.; CHALLA CáM N. Real-time and secure wireless health monitoring. int. j. telemedicine appl. *Computer*, p. 1–10, 2008.
- PROTÉGÉ. *Overview*. Dezembro 2009. Acessado em Setembro de 2008. Disponível em: <<http://protege.stanford.edu/overview/>>.

PÚBLICO, S. *Portal do Software Público Brasileiro*. Setembro 2009. Acessado em Novembro de 2009. Disponível em: <<http://www.softwarepublico.gov.br/>>.

RCASOFT. *JME documentation*. Janeiro 2008. Acessado em Maio de 2009. Disponível em: <<http://www.rcasoft.com.br>>.

SANT'ANNA, F. *Desenvolvimento de Aplicações Declarativas para TV Digital no Middleware Ginga com Objetos Imperativos NCLua*. Fortaleza, CE, Brasil: WebMedia, 2009. CD ROM v3. ISSN 2175-9669.

SMSLIB. *Programmer's Library for Sending and Receiving SMS*. Outubro 2008. Acessado em Novembro de 2008. Disponível em: <<http://smslib.org/>>.

SOARES, L. F. G.; RODRIGUES, R. F.; MORENO, M. F. Declarative environment of the brazilian digital tv system. *Journal of the Brazilian Computer Society*, v. 13, n. 1, p. 37–46, March 2007.

SOMMERVILLE, I. *Engenharia de Software*. São Paulo, SP, Brasil: PEARSON, 2007. Hardcover. ISBN 978-85-88639-28-7.

VALERO, M.; L.VADILLO. An implementation framework for smart home telecare services. *Future Generation Communication and Networking (FCGN)*, v. 2, p. 60–65, Dezembro 2007.

VALERO, M. et al. An implementation framework for smart home telecare services. *Future Generation Communication and Networking*, IEEE Computer Society, Los Alamitos, CA, USA, v. 2, p. 60–65, 2007.

VIANA, N. S. *Integração entre o Middleware Brasileiro de TV Digital e Serviços de Dispositivos Eletrônicos em Redes OSGI*. Dissertação (Mestrado) — Universidade do Amazonas, 2009.

VUGT, S. van. *Beginning Ubuntu Server Administration*. USA: Apress, 2007. Hardcover. ISBN 9781590599235.

Apêndice A

Publicações

- SILVA, Vandermi; MAIA O.B; MARTINIANO A.L; VIANA Nairon S.; LUCENA Vicente F. Jr. Cenário de Convergência entre a TV digital e Dispositivos Eletrônicos em Sistemas de Automação Residencial. CONNEPI-2008, Fortaleza -CE
- SILVA, Vandermi; MAIA O.B; MARTINIANO A. L; BINDÁ Jomara; LUCENA Vicente F. Jr. Implementação de Healthcare Integrando um Gateway e a TV digital para Monitorar Idosos. CONNEPI-2009, Belém -PA
- SILVA, Vandermi; LUCENA Vicente F. Jr. Monitoring of Hypertensive Patients through Conventional Medical Devices Integrated to the Brazilian Digital TV. EuroITV'10, June 9–11, 2010, Tampere, Finland.
- SILVA, Vandermi; LUCENA Vicente F. Jr. mHealth- Integrating Conventional Medical Devices with the Brazilian Digital TV and Mobile Phones 12th IEEE International Conference on E-Health Networking, Applications and Services (IEEE Healthcom) July 1-3, 2010, Lyon, France.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)