



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Manuella Dias Carvalho Silva

**MONITORAMENTO REMOTO PREVENTIVO DE PACIENTES COM
DOENÇAS CARDIOVASCULARES UTILIZANDO DISPOSITIVO
MÓVEL COMO AGENTE INTELIGENTE**

João Pessoa
Maio de 2012

Manuella Dias Carvalho Silva

**MONITORAMENTO REMOTO PREVENTIVO DE PACIENTES COM
DOENÇAS CARDIOVASCULARES UTILIZANDO DISPOSITIVO
MÓVEL COMO AGENTE INTELIGENTE**

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informática da
Universidade Federal da Paraíba, como requisito
parcial para a obtenção do Título de Mestre em
Informática.

Linha de Pesquisa: Computação Distribuída

Orientador: Dr. Claurton de Albuquerque Siebra

João Pessoa

Maio de 2012

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca Central da Universidade Federal da Paraíba

S586m Silva, Manuella Dias Carvalho.

Monitoramento remoto preventivo de pacientes com doenças cardiovasculares utilizando dispositivo móvel como agente inteligente / Manuella Dias Carvalho Silva.-- João Pessoa, 2012.

140f. : il.

Orientador: Clauirton de Albuquerque Siebra

Dissertação (Mestrado) – UFPB/CCEN

1. Informática. 2. Sistemas de Monitoramento Remoto.
3. Prevenção da saúde. 4. Teleassistência. 5. Sistemas pervasivos.
6. Sistemas inteligentes.

UFPB/BC

CDU: 004(043)

1
2

Ata da Sessão Pública de Defesa de Dissertação de
Mestrado de **MANUELLA DIAS CARVALHO
SILVA**, candidato ao Título de Mestre em
Informática na Área de Sistemas de Computação,
realizada em 24 de MAIO de 2012.

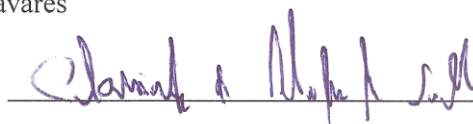
3
4

5 Aos vinte e quatro dias do mês de maio do ano dois mil e doze, às quatorze horas, no
6 Auditório do CCEN - da Universidade Federal da Paraíba, reuniram-se os membros da
7 Banca Examinadora constituída para examinar a candidata ao grau de Mestre em
8 Informática, na área de “*Sistemas de Computação*”, na linha de pesquisa “*Computação*
9 *Distribuída*”, a Srta. **MANUELLA DIAS CARVALHO SILVA**. A comissão
10 examinadora foi composta pelos professores doutores: CLAUIRTON DE
11 ALBUQUERQUE SIEBRA (PPGI-UFPB), Orientador e Presidente da Banca
12 Examinadora, NATASHA CORREIA QUEIROZ LINO (DI-UFPB), como examinador
13 interno e CRISTINE MARTINS GOMES DE GUSMÃO (UFPE), como examinador
14 externo. Dando início aos trabalhos, o professor CLAUIRTON DE ALBUQUERQUE
15 SIEBRA, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e
16 passou a palavra ao candidato para que o mesmo fizesse, oralmente, a exposição do
17 trabalho de dissertação intitulado “MONITORAMENTO REMOTO PREVENTIVO DE
18 PACIENTES COM DOENÇAS CARDIOVASCULARES UTILIZANDO DISPOSITIVO
19 MÓVEL COMO AGENTE INTELIGENTE”. Concluída a exposição, o candidato foi
20 argüido pela Banca Examinadora que emitiu o seguinte parecer: “aprovado”. Assim sendo,
21 deve a Universidade Federal da Paraíba expedir o respectivo diploma de Mestre em
22 Informática na forma da lei e, para constar, eu, professora Tatiana Aires Tavares,
23 Coordenadora deste Programa, servindo de secretária lavrei a presente ata que vai
24 assinada por mim mesmo e pelos membros da Banca Examinadora. João Pessoa, 24 de
25 maio de 2012.

26
27


Tatiana Aires Tavares

Prof. Dr. Clauriton de Albuquerque Siebra
Orientador (PPGI-UFPB)

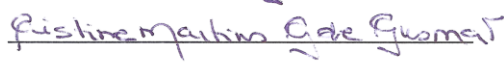


Profa. Dra. Natasha Correia Queiroz Lino



Examinador Interno (DI-UFPB)

Profa. Dra. Cristine Martins Gomes De Gusmão
Examinador Externo (UFPE)



28

Ao grande criador da vida e
aos meus grandes amores que me inspiram a
alcançar todos os meus sonhos.

Agradecimentos

E eis que já se passaram mais dois anos e dois meses... Chega o fim de mais uma etapa da minha vida. Foram lutas, estudos, pesquisas, aprendizados, oportunidades, amadurecimentos, conquistas diversas, que sobre incentivos, paciência, correções e carinho me permitiram o alcance desta vitória. Por isso, então, aproveito este espaço para prestar agradecimentos àqueles que estiveram ao meu lado e me ajudaram a construir cada instante que me permitiu chegar até aqui.

Primeiramente, agradeço àquele que sempre está ao meu lado, mesmo sem que eu perceba. Àquele que me dá força, paciência, sabedoria, amor e fé, que confia em mim e me apoia em tudo, que me dá todas as oportunidades e a liberdade de escolha. Muito obrigada Deus por me dar este dom tão maravilhoso que é a vida.

Aos meus pais por me ensinarem todas as coisas que me guiaram pela vida, me conduziram a ser quem sou. Por depositar tamanha fé e confiança em mim, por me ensinarem a ser livre, independente, forte, a cativar e cultivar a vida, a amar os que estiverem ao meu redor, e a aprender que errar faz parte do crescimento. Por todas as oportunidades que me concederam e, principalmente, pelo exemplo de filho e filha, pai e mãe, indivíduos que vocês foram, são e sempre serão para mim. MUITÍSSIMO obrigada Renilda Dias C. da Silva e Manoel Sérgio da Silva (*in memoriam*)!

Ao meu querido irmão, Sérgio Dias C. Silva, por acreditar em mim, por ter paciência e cuidado comigo. Por me ajudar a pensar em soluções para meus problemas, debater comigo sobre as questões da vida e por me inspirar a seguir meus sonhos mesmo quando eu os esqueci. Sou eternamente grata por tudo.

Ao meu namorado Lucas C. Cruz por todo carinho, empenho, paciência, incentivo, atenção, acolhimento e principalmente por esse amor que me ajuda a construir e conquistar tantos sonhos. Por escutar meus segredos, por dividir sua vida comigo, por tudo que me ensinou, por estar sempre ao meu lado. Pelos respaldos que me deu para que eu pudesse concluir este trabalho, me dando seu computador, restaurando meus arquivos perdidos, por discutir e solucionar comigo problemas de implementação e por suas correções textuais. A você meu lindo amor, sou muito grata.

A toda minha família pelo apoio oferecido, pela paciência frente às minhas ausências, pela confiança, amor, carinho e pelos inúmeros conhecimentos que me proporcionaram. Em especial aos meus avós que fizeram essa família linda e nos ensinaram a termos essa união. Agradeço a vocês todos por tudo.

Obrigada a todos os meus amigos presentes ou ausentes, que passaram ou estão na minha vida e que sempre me dão esperança para continuar, me incentivam a sonhar, compartilham suas vidas comigo, me auxiliam nos estudos, me dão apoio emocional, me apresentam oportunidades e acreditam em mim. Obrigada, principalmente: aos meus “quartetos fantásticos” que mesmo de áreas tão distintas da minha sempre acompanham minhas investidas: Carolina Raissa, Klaus, Alysson, Felipe, Priscilla, Larissa, Beatriz, Paula e Raissa; aqueles a quem denomino de irmãos que conhecem e sabem o que estou passando ou sentindo mesmo sem estarem ao meu lado; a todos da Fábrica 61, uma turma de sonhadores cultivada a partir do departamento de Informática (DI) da UFPB que sempre alcança diversas conquistas; aos meus alunos do estágio docência e das disciplinas lecionadas no Campus I da UEPB que me ensinaram tanto com suas atitudes e com suas vidas; aos que têm me recebido tão bem nos novos caminhos que tomo para a próxima etapa de minha vida, em especial a turma 2012.1 de Direito da UFPB, Campus I.

Agradeço a todos aqueles que passaram por minha vida ao longo deste período do mestrado, por me fornecerem sorrisos, apoio, críticas, inspirações, limites, por contribuírem tão ricamente para minha formação. Muito obrigada a todo o DI da UFPB e ao Departamento de Computação e de Matemática da UEPB por me auxiliarem em minha formação acadêmica e pessoal.

Um obrigado especial àqueles professores e pesquisadores que me auxiliaram com materiais, ideias, críticas e sugestões ao longo da elaboração deste trabalho, em especial a Dra. Tatiana Aires pela sua disponibilidade e a Dra. Natasha Lino, Dr. Gustavo Motta sempre dispostos a auxiliar seus alunos e por suas leituras do trabalho e a Dra. Cristine Gusmão por sua leitura e disponibilidade ao exame deste trabalho.

Agradeço de maneira muito especial ao Dr. Claurton de Albuquerque Siebra pela oportunidade oferecida, pela confiança depositada, por todo o cuidado, atenção e por todas as sugestões para desenvolvimento deste trabalho. Por sua dedicação e paciência, por seus conselhos e sua amizade, por acreditar em mim mesmo quando eu não o fazia. Muitíssimo obrigada professor pela sua orientação.

Enfim, a todos aqueles aqui citados ou não que passaram pela minha vida nesses vinte e dois anos, em especial nesses dois últimos anos, contribuíram para que aqui eu chegasse, então: muito obrigada!

“O futuro pertence aqueles que acreditam na
beleza dos seus sonhos.”

Eleanor Roosevelt

Resumo

Vários governos e países têm demonstrado bastante interesse em definir políticas públicas para reduzir os gastos com os serviços de saúde. Dessa maneira, diversos pesquisadores têm procurado desenvolver soluções que reduzam tais custos. Entre essas soluções, estão os sistemas de monitoramento remoto do estado de saúde de pacientes críticos, aqueles que sofrem de doenças cardíacas, doenças crônicas ou idosos que precisam de atendimento contínuo. Esses sistemas permitem a identificação de situações de risco e a adoção de procedimentos preventivos, o que reduz os gastos com saúde. Além disso, eles permitem uma melhoria na qualidade e no tempo de vida de seus usuários. Várias são as abordagens para construção desses sistemas. De maneira geral, elas se dividem em dois grandes grupos: os *home care* e os móveis. A abordagem utilizada nesse trabalho defende a utilização de sistemas de monitoramento móveis pode ser mais eficiente, uma vez que o acompanhamento da saúde será realizado independentemente da localidade ou da movimentação do paciente. O trabalho aqui apresentado faz uso de dispositivos pervasivos para monitoramento do estado de saúde de pacientes com doenças cardiovasculares, que são a maior causa de morte no mundo. Nesse sentido, esta dissertação contribui com um aplicativo para uso em telefones móveis que atue como um agente inteligente personalizado para o monitoramento preventivo de doenças cardiovasculares, utilizando um mecanismo de inteligência baseado em regras de produção. Esse aplicativo, que recebeu o nome de *MonitorPrevIntel*, com base na leitura contínua de cinco sinais fisiológicos, detecta situações anormais, de perigo para a saúde de seu usuário, e emite procedimentos de alarme. Para detectar as situações anômalas o aplicativo utiliza um mecanismo de inteligência totalmente embutido no telefone celular, com base de conhecimento personalizada para cada usuário. Os alarmes emitidos pelo sistema avisam ao usuário sobre a detecção de situações anormais de saúde e avisam, através de mensagens SMS, ou acionam, através de chamada telefônica, uma central médica sobre essa situação. O funcionamento do *MonitorPrevIntel* é controlado por seu mecanismo de inteligência, com base no estado de saúde do usuário, permitindo que o mesmo execute de maneira autônoma, independentemente da ativação pelo usuário.

Palavras chave: Teleassistência; Sistemas pervasivos; Sistemas inteligentes; Prevenção da saúde.

Abstract

Several Governments and countries have shown great interest in defining public policies to reduce spending on health services. In this way, several researchers have sought to develop solutions that reduce such costs. Among these solutions are the remote monitoring systems of the State of health of critical patients, those suffering from heart disease, chronic diseases or elderly who need continuous care. These systems allow the identification of risk situations and the adoption of preventive procedures, reducing health spending. In addition, they allow an improvement in quality and lifetime of your users. Several approaches to the construction of these systems. Generally, they are divided into two main groups: the home care and the furniture. The approach used in this paper advocates the use of mobile monitoring systems can be more efficient, since the health monitoring will be carried out regardless of locale or patient drive. The work presented here makes use of pervasive devices for monitoring of the health status of patients with cardiovascular diseases, which are the biggest cause of death in the world. In this sense, this dissertation contributes with an application for use in mobile phones that act as an intelligent agent to the preventive monitoring of cardiovascular diseases, using a rules-based intelligence engine. This application, which has received the name of *MonitorPrevIntel*, based on the continuous reading of five physiological signals, detect abnormal situations of danger for the health of your server, and emits alarm procedures. To detect anomalous situations the application uses a mechanism fully embedded intelligence on the mobile phone, with custom knowledge base for each user. The alarms emitted by the system warn the user about the detection of abnormal situations and advise, via SMS messages, or trigger, via phone call, a medical center about this situation. The *MonitorPrevIntel* operation is controlled by your intelligence mechanism, based on the State of health of the user, allowing the same run autonomously, independently of the user activation.

Keywords: Telemonitoring; Pervasive systems in health care; Intelligent systems in health care; Health prevention.

Lista de Figuras

Figura 2.1 - Pilha de protocolos ZigBee.....	28
Figura 2.2 - Topologias das redes ZigBee.....	29
Figura 2.3 - Pilha de protocolos do Bluetooth x Modelo OSI.....	31
Figura 2.4 - <i>Scatternet</i> Bluetooth com seis redes.....	32
Figura 2.5 - Arquitetura SMS.....	35
Figura 2.6 - Esquema de um <i>Protocol Data Unit</i>	36
Figura 2.7 - Algumas definições para IA, agrupadas em quatro categorias (RUSSEL; NORVIG, 2003, p. 2)	38
Figura 2.8 - Esquema de interação de um agente inteligente com o ambiente.....	39
Figura 2.9 - Comparação modelo Newell e modelo de SE.	41
Figura 2.10 - Funcionamento do desenvolvimento de uma aplicação KEOPS (ALBUQUERQUE, 2002, p. 112).....	43
Figura 3.1 - Arquitetura do sistema proposto por JAFARI et al (2007).....	45
Figura 3.2 - Interação no sistema proposto por ANDREÃO, PEREIRA FILHO e CALVI (2006).	46
Figura 3.3 - Protótipo AMON (ANLIKER <i>et al.</i> , 2004).	47
Figura 3.4 - Visão geral do sistema proposto por ANLIKER <i>et al.</i> (2004).....	47
Figura 3.5 - Arquitetura do sistema desenvolvido por MACHADO <i>et al.</i> (2008).	48
Figura 3.6 - Arquitetura do sistema desenvolvido por LEE <i>et al.</i> (2006).	49
Figura 3.7 – Diagrama de fluxo de informações entre os diferentes atores de um sistema médico, organizado por LIEBERMAN e MASON (2002).	50
Figura 3.8 - Integração entre JADE e Agilla proposto por HERBERT <i>et al.</i> (2006).....	51
Figura 3.9 - Arquitetura do sistema desenvolvido por COPETTI (2010).	51
Figura 4.1 - Modelo Espiral Completo (SOMMERVILLE, 2007).	55
Figura 4.2 - Arquitetura da plataforma JME.	58
Figura 4.3 - Ciclo de vida de um MIDlet.	60
Figura 5.1 - Visão geral da arquitetura do sistema <i>MonitorPrevIntel</i>	63
Figura 5.2 - Base de conhecimentos especificada na linguagem definida neste trabalho.	66
Figura 5.3 - Diagrama de pacotes do sistema.	68
Figura 5.4 - Diagrama de classes do <i>MonitorPrevIntel</i>	69
Figura 5.5 - Interface <i>TokenConstants</i>	70
Figura 5.6 - Caso de uso do <i>MonitorPrevIntel</i>	74

Figura 5.7 - Diagrama de atividades do método <i>fireRuleBase()</i> da classe <i>AgentMidlet</i>	76
Figura 5.8 - Diagrama de atividades do método <i>action()</i> da classe <i>AgentMidlet</i>	78
Figura 5.9 - Modelo (a) e exemplo (b) de SMS enviada pelo aplicativo informando uma situação de anormalidade no sinais vitais.	80
Figura 6.1 - Interface gráfica do WMA da ferramenta Wireless Toolkit 2.5.2.	83
Figura 6.2 - Exemplo de regra de produção do formato apresentado em Copetti (2010).	84
Figura 6.3 - Telas do aplicativo: (a) exibindo as mensagens iniciais, (b) após a execução do aplicativo com a medição 1 da Tabela 6 e o usuário em repouso.	87
Figura 6.4 - Telas do aplicativo: (a) questionando o usuário sobre sua atividade e (b) exibindo uma mensagem de emergência.	88
Figura 6.5 - Telas do aplicativo: (a) exibindo mensagens após execução de ação de emergência, (b) após execução de ação alerta seguida de envio de backup da base de dados.	89

Lista de Tabelas

Tabela 2.1 - Comparativo entre Classic Bluetooth, Bluetooth Low Energy e ZigBee.	34
Tabela 2.2 - Exemplo de codificação de conhecimento em regra de produção.	42
Tabela 3.1 - Síntese das características dos sistemas encontrados no estado da arte.	53
Tabela 4.1 - Requisitos mínimos para os dispositivos MIDP.	58
Tabela 5.1 - Elementos da linguagem de especificação da base de conhecimentos.	67
Tabela 6.1 - Dados sintéticos que representam oito medições.	86
Tabela 7.1 - Síntese das características das principais soluções de sistemas de monitoramento remoto preventivo.	94
Tabela A.1 - Quadro de referência para a classificação do desvio da pressão arterial sistólica.	106
Tabela A.2 - Intervalos para classificação da frequência cardíaca.	107
Tabela C.1 - Regras utilizadas para teste do MonitorPrevIntel.	109
Tabela C.1 - Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.	112

Lista de Abreviaturas

3GPP	Third Generation Partnership Project
AES	Advanced Encryption Standard
AHF	Adaptive Hopping Frequency
AMON	Advanced care and alert portable telemedical MONitor
AMS	Application Management Software
ANATEL	Agência Nacional de Telecomunicações
API	Application Programming Interface
BLE	Bluetooth Low Energy
BS	Base Stations
CBC- MA	Cipher Block Chaining Message Authentication Code
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
CM	Central de Monitoramento
DMS	Data Management System
ECDH	Elliptic Curve Diffie Hellman
ECG	Eletrocardiograma
EOF	End Of File
ETSI	European Telecommunications Standards Institute
FDMA	Frequency Division Multiple Access
GBD	Global Burden of Disease
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HS	High Speed
IA	Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific and Medical
JAD	Java Application Descriptor
JAR	Java ARchive file
JCE	Java Cryptography Extension
JCP	Java Community Process
JDK	Java Development Kit

JEOPS	Java Embedded Objects Production Systems
JME	Java Mobile Edition
JNI	Java Native Interface
JVM	Java Virtual Machine
KB	Knowledge Base
KEOPS	k Embedded Objects Production Systems
KVM	kilobyte Virtual Machine
MAC	Media Access Control
MI	Módulo de Inferência
MIDP	Mobile Information Device Profile
MR	Módulo Relator
NWK	Network
OTA	Over The Air
PC	Power Computer
PDA	Personal Digital Sssistant
PDU	Protocol Data Unit
PHY	Physical
RMS	Record Management System
RMS	Record Management System
RSC	Redes de Sensores Corporais
SAD	Sistemas de Apoio à Decisão
SE	Sistemas Especialistas
SIG	Special Inerest Group
SME	Short Message Entities
SME-o	SME originadora
SME-t	SME terminadora
SMS	Short Message Service
SMSC	SMS Center
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TIA	Telecommunications Industry Association
TIC	Tecnologia da Informação e Comunicação
TP-DCS	Transfer Protocol Data Coding Scheme

TPDU	Transfer Protocol Data Unit
TP-MMS	Transfer Protocol More Messages to Send
TP-MTI	Transfer Protocol Message Type Indicator
TP-OA	Transfer Protocol Originating Address
TP-PI	Transfer Protocol Protocol Identifier
TP-RP	Transfer Protocol Reply Path
TP-SCTS	Transfer Protocol Service Centre Time Stamp
TP-UD	Transfer Protocol User Data
TP-UDHI	Transfer Protocol User Data Header Indicator
TP-UDL	Transfer Protocol User Data Length
UML	Unified Modeling Language
UP	Unidades de Processamento
UR	Unidade Remota
UTF-16	16-bit Unicode Transformation Format
WHO	World Health Organization
WMA	Wireless Messaging API
WPAN	Wireless Personal Area Network
WSC	Wireless Switching Center
WTK	Wireless Toolkit
XML	Extensible Markup Language

Sumário

1 Introdução	18
1.1 Motivação.....	20
1.2 Objetivos.....	21
1.3 Justificativas.....	22
1.4 Estrutura do Trabalho.....	24
2 Tópicos Relevantes para o Sistema	26
2.1 Tecnologias para Transmissão Sem Fio	26
2.1.1 Tecnologia ZigBee	27
2.1.2 Tecnologia Bluetooth	30
2.2 Short Message Service (SMS)	35
2.3 Inteligência Artificial	38
3 Estado da Arte	44
4 Procedimentos Metodológicos	54
4.1 Modelagem.....	56
4.2 Implementação	56
5 MonitorPrevIntel	61
5.1 Características Gerais	61
5.2 Arquitetura	62
5.2.1 Sistema de Produção.....	63
5.2.2 Classes do MonitorPrevIntel	68
5.2.3 Base de Dados	79
5.3 Funcionamento.....	79
6 Testes e Resultados	82
7 Considerações Finais	91
7.1 Discussões.....	93

7.2 <i>Trabalhos Futuros</i>	97
Referências	99
Apêndice A – Classificação das variações dos Sinais Vitais	106
Apêndice B – Base de Conhecimentos do Caso de Teste	108
Apêndice C – Testes	109
Anexo A – Regras de Produção Utilizadas para Teste do MonitorPrevIntel	140

Capítulo

1

Introdução

“A sabedoria começa na reflexão. O início da sabedoria é a admissão da própria ignorância. Todo o meu saber consiste em saber que nada sei.”

Sócrates

O crescente aumento da população mundial chama a atenção de vários governos ao redor do mundo para conduzir seus esforços no planejamento adequado dos serviços de saúde, principalmente nos países em desenvolvimento e onde seus habitantes estejam distribuídos ao longo de várias regiões. Além disso, cresce o número de doentes crônicos, aqueles que precisam de longos e detalhados períodos de observação (COPETTI, 2010). Infelizmente, a indisponibilidade de leitos em hospitais faz com que muitos acabem por ter o tempo de vida reduzido, em especial os pacientes com doenças cardiovasculares.

As doenças cardiovasculares são aquelas que afetam o coração e/ou os vasos sanguíneos, tais como doenças coronárias, doenças cerebrovasculares, pressão arterial elevada, doença arterial periférica, doença cardíaca reumática, cardiopatia congênita e insuficiência cardíaca. Com base na análise da WHO *Global Burden of Disease* (GBD) de fevereiro de 2009, as doenças cardiovasculares são as principais causas de morte no mundo, levando ao óbito cerca de 17,1 milhões de pessoas por ano (ESTIMATED, 2004). Dentre estas mortes, 80% são pessoas de classe média ou baixa (FACTSHEET, 2011), exatamente as que mais sofrem em decorrência da falta de leitos, medicamentos e cuidados médicos em geral.

Esta alta demanda mostra a necessidade de uma reestruturação na infraestrutura de serviços de saúde, principalmente no âmbito da saúde pública. Segundo Barbosa *et al.* (2004), quatro paradigmas têm norteado este novo modelo de serviço, que são: centralizar os serviços de saúde no paciente, descentralizar a prestação de serviços levando-o para prestação de

serviços domiciliares, priorização da qualidade de vida dos pacientes e na prevenção de doenças com base em monitoramento de hábitos de vida.

Uma possível solução para estes problemas, que reúne estes quatro paradigmas, é a utilização de técnicas de monitoramento remoto do estado de saúde do paciente. Esta solução possibilita o acompanhamento dos estados fisiológicos do paciente e uma posição preventiva em relação às doenças que possam atingi-lo. Esses sistemas possibilitam algumas vantagens, tais como a redução dos custos com cuidados de saúde, visto que muitas doenças podem ser prevenidas a tempo de evitar doenças e internações; eficiência, à medida que estas técnicas permitem o acompanhamento do estado fisiológico dos pacientes continuamente além de executar algoritmos de busca por situações de perigo à saúde destes e melhoria na qualidade de vida do paciente que pode sentir-se mais seguro ao ter sua saúde monitorada diuturnamente e sem o desconforto e o risco das infecções por estar em um quarto de hospital.

Diante de tudo isso, estas técnicas estão tornando-se comuns e mostram-se eficazes na identificação de situações de risco e na adoção de procedimentos de saúde preventivos, bem como auxiliando no acompanhamento do quadro de evolução clínica de pacientes. Diversos estudiosos têm pesquisado e desenvolvido diferentes abordagens para o uso de técnicas de monitoramento remoto da saúde.

De maneira geral, estas pesquisas levam a modelos nos quais os pacientes são monitorados em suas casas, *home care*, a partir de dispositivos captadores de sinais vitais que contenham algum mecanismo de comunicação com uma central médica¹, na qual especialistas executam ações efetivas em relação aos cuidados de sua saúde. Infelizmente os altos custos para manutenção destes sistemas e a forma evasiva com a qual eles entram na rotina dos pacientes torna, quase sempre, inviável a sua utilização de forma eficaz.

Este trabalho tem o intuito de investigar técnicas para desenvolvimento de um sistema de monitoramento que possa reduzir os custos com os serviços de saúde e aumentar a qualidade de vida dos pacientes com doenças cardiovasculares. Em adição, pretende-se construir um sistema de baixo custo que sirva de base para o monitoramento remoto preventivo de maneira pervasiva, flexível e inteligente.

¹ O termo central médica se refere a qualquer centro de atendimento à saúde, seja um hospital, um posto de saúde, uma clínica especializada, uma central de emergência independente ou até um consultório independente.

1.1 Motivação

Uma alternativa para a solução do monitoramento remoto das condições de saúde dos pacientes é a adoção de tecnologias ubíquas². Essas poderiam permitir o acompanhamento diário dos seus usuários³ auxiliando na prevenção e na detecção de doenças ou situações de risco de morte, tal como um princípio de infarto, sem alterar as atividades cotidianas dos pacientes usuários do sistema.

Um dispositivo que se apresenta como uma alternativa bastante acessível para o monitoramento remoto pervasivo é o telefone móvel, pois é um eletrônico de uso comum e com grande abrangência entre a população, com preço em queda ano após ano e com tecnologias cada vez mais sofisticadas. No Brasil, segundo a ANATEL, 99,8% da população é atendida por pelo menos uma prestadora de serviços de telefonia móvel (COBERTURA, 2011) e a teledensidade, indicador usado para mostrar o número de acessos à telefonia móvel para cada grupo de 100 habitantes (NÚMERO, 2011), chegou à marca de 123,87 (BRASIL, 2012).

Os dispositivos de tecnologia móvel contêm inúmeras soluções de conectividade que permitem acesso às redes de informação disponíveis a qualquer instante ou em qualquer lugar (WOCHNER, 2005), bem como permitem o desenvolvimento de sistemas mais sofisticados, por exemplo, com o uso de técnicas de inteligência artificial. Esses sistemas podem ser construídos para analisar os diversos cenários nos quais o paciente está inserido e liberar sinais de alertas e notificações sobre seu estado.

Um monitoramento adequado pode ser alcançado através do uso de dispositivos com sensores sem fio posicionados no corpo dos pacientes, que irão aferir e transmitir os dados fisiológicos diuturnamente. Esses dados serão coletados por um dispositivo computacional que verifique, armazene e/ou processe os sinais recebidos. O processamento destes sinais pode ir desde um serviço de retransmissão dos dados do paciente para uma central médica até um sistema mais complexo que analise os dados, identifique situações, registrem/liberem

² O termo tecnologia ubíqua possibilita o uso de inteligência na percepção do ambiente e adequação do sistema ao usuário, incorporando os conceitos de: computação móvel que remonta a movimentação de serviços computacionais com os usuários independentemente de sua localização e de computação pervasiva que implica na presença de dispositivos computacionais embarcados no ambiente de forma imperceptível (ARAUJO, 2003).

³ A palavra usuário será utilizada diversas vezes ao longo do trabalho como referência ao paciente com doença cardiovascular que faz uso de algum sistema de monitoramento de saúde.

notificações e comunique-se com o paciente, o médico ou qualquer outro sistema computacional.

Para os profissionais de saúde, estes sistemas de monitoramento permitem o acompanhamento de maneira remota do estado dos pacientes em seu cotidiano e a evolução de seu tratamento, reajustando-o de maneira adequada (CARVALHO *et al.*, 2010). Para o paciente, isso diminuirá a necessidade de internação hospitalar e a frequência de visitas aos consultórios médicos e possibilitar uma melhor qualidade de vida. Os sistemas bem elaborados conseguem, em resumo, reduzir significativamente o custo dos serviços de saúde, seja para o governo, para os planos de saúde ou para os pacientes, bem como aumentar a eficácia do diagnóstico e do tratamento médico.

1.2 Objetivos

Dentro do contexto apresentado, este trabalho apresenta a concepção e o desenvolvimento de um aplicativo inteligente para celular que auxilie no processo de monitoramento remoto de sinais vitais de pacientes com doenças cardiovasculares. O principal objetivo é a criação e construção de um aplicativo para dispositivos móveis que atue como agente assistente personalizado para monitoramento remoto preventivo de doenças cardiovasculares.

O aplicativo inteligente será responsável por verificar os sinais vitais do usuário, segundo parâmetros pré-definidos de normalidade e regras de produção estabelecidas por profissionais de saúde, e de ativar e gerenciar ações de armazenamento de dados, interação com o usuário, envio de mensagens de texto e inicialização de chamadas telefônicas. Os objetivos secundários, necessários à construção deste sistema, são:

- Analisar sistemas de monitoramento remoto de pacientes existentes no estado da arte, que utilizem ou não inteligência artificial e telefones móveis;
- Investigar os tipos de tecnologias para desenvolvimento de aplicativos para dispositivo móvel;
- Definir uma estrutura que suporte o monitoramento remoto de pessoas com doenças cardiovasculares;

-
- Implementar um sistema que seja capaz de ler uma base de conhecimento⁴ composta por regras de produção personalizadas para cada paciente e inferir sobre a mesma;
 - Implementar um sistema de envio de mensagens de texto cifradas, através do celular;
 - Utilização de um conjunto de regras de produção já existente, uma vez que não poderemos contar no projeto com especialistas da área de saúde, para gerar diferentes bases de conhecimento, com a finalidade de validar o funcionamento do aplicativo.

1.3 Justificativas

Segundo ANDREÃO, PEREIRA FILHO e CALVI (2006), o monitoramento diuturno dos pacientes com doenças cardiovasculares, de maneira particular, melhora a qualidade da internação de pacientes domiciliares, reduz os números de leitos ocupados, os gastos com internações e auxilia na identificação precoce de doenças, fazendo com que o tratamento seja mais rápido e eficaz. Infelizmente, a maioria das pesquisas atuais, referentes ao monitoramento remoto de pacientes, está focada em:

- Discutir aspectos relacionados à infraestrutura dos sistemas, principalmente no que se refere a quais padrões e tecnologias de rede utilizar e quais protocolos ou mecanismos utilizarem para manter a informação segura (ANDREÃO *et al.*, 2006; XIAO; CHEN, 2008);
- Utilizar dispositivos móveis como receptores e transmissores (roteadores) de informações obtidas através dos sensores instalados no paciente para envio a base central de processamento, possibilitando o monitoramento do paciente independentemente da área onde o usuário esteja, porém não utilizando o celular na análise dos sinais fisiológicos (LEE *et al.*, 2006; JAFARI *et al.*, 2007; ANLIKER *et al.*, 2004);

⁴ O conceito de base de conhecimentos será apresentado na seção 2.3 Inteligência Artificial, do Capítulo 2.

- Criar ambientes *home care*, no qual o monitoramento é feito na casa do paciente, onde os domicílios tornam-se uma espécie de extensão do ambiente hospitalar que têm um canal para que os dados trafeguem até uma base central para processamento das informações (COPETTI, 2010; ANDREÃO *et al.*, 2006).

Este trabalho defende que ainda há muito a ser explorado para a melhoria das maneiras de acompanhar a saúde de pacientes remotamente. Avançando sobre os sistemas tradicionais de tratamento remoto preventivo de saúde, presentes no estado da arte, à medida que defende a união de aspectos da teleassistência com elementos da inteligência artificial para prover ao paciente um sistema de monitoramento móvel confiável⁵. Este também deve possibilitar aos seus usuários a experiência de uso de um sistema de acompanhamento preventivo de saúde que se incorpore e se adapte ao seu dia-a-dia sem interferir no seu conforto ou em sua rotina, fazendo com que esqueçam o desconforto causado pela extensão hospitalar em sua casa que quase sempre é provocada pelos sistemas *home care*.

O aparelho celular é um dispositivo que a cada dia torna-se mais sofisticado em tecnologia e a um preço cada vez mais baixo. Os recursos tecnológicos dos telefones móveis permitem que eles próprios ajam como entidades autônomas capazes de tomar diferentes decisões para o monitoramento de sinais vitais de maneira inteligente e ainda sirvam como canal de comunicação confiável, visto que a cobertura da telefonia móvel alcança quase 100% do território brasileiro (COBERTURA, 2011). Por fazerem parte do cotidiano das pessoas de maneira pervasiva, visto que elas os carregam a qualquer lugar que vão, permitem que o acompanhamento à saúde seja feito independente de sua localização física.

Neste trabalho, o telefone celular é apresentado como um dispositivo computacional capaz de exercer a função de um agente assistente personalizado automático, através de um sistema embutido capaz de receber sinais vitais dos pacientes, fazer o processamento e a análise dos dados, além de:

- Perceber situações de anormalidades com os dados fisiológicos do paciente, a partir das regras de produção⁶ específicas para ele;
- Emitir mensagens aos usuários, questionando-os sobre alguma atividade ou exibindo alarmes sobre situações de alerta e procedimentos;
- Emitir mensagens de texto alertando sobre alguma situação crítica do paciente;

⁵ Confiável porque, além de aferir os sinais vitais do paciente e armazená-los, o sistema será capaz de realizar processamentos locais a partir de mecanismos inteligentes.

⁶ O conceito de regras de produção será explicado na seção 2.3 Inteligência Artificial, do Capítulo 2.

- Inicializar chamadas telefônicas para centrais de atendimento em situações críticas de saúde;
- Emitir relatórios dos dados fisiológicos dos pacientes.

Além destas características, este sistema se propõe a ser uma aplicação de monitoramento automático, que opera sem necessitar que o usuário o acione. A utilização de inteligência local ao dispositivo que é responsável pelo monitoramento também é um avanço considerável no estado da arte. Este uso de inteligência artificial embarcada no dispositivo monitor já começou a ser desenvolvido para sistemas *home care*, como em COPETTI (2010). Porém, nada foi encontrado no estado da arte sobre o uso de um motor de inferência local a um telefone móvel.

Uma discussão sempre pertinente quando se trata do uso das redes de telefonia móvel para transmissão de dados sigilosos, como é o caso de dados médicos, é a forma insegura como estes dados trafegam de um emissor até um receptor. Para evitar problemas de vazamento de dados, o sistema conta com um módulo de criptografia que cifra todas as mensagens de texto enviadas por ele.

Para alcançar os objetivos propostos, a contribuição deste trabalho é o desenvolvimento de um sistema móvel e inteligente de monitoramento remoto preventivo, chamado *MonitorPrevIntel*⁷ ou de maneira extensa *Monitor Intelligent Prevention*, que permita o acompanhamento da saúde de seu usuário de forma ubíqua e confiável. Espera-se que este sistema beneficie o cotidiano de pacientes com doenças cardiovasculares, os quais necessitem de acompanhamento de sua saúde, melhorando sua qualidade de vida e os custos em tratamentos de saúde.

1.4 Estrutura do Trabalho

Este trabalho está estruturado em sete capítulos. O primeiro que aqui se encerra trouxe uma visão geral do trabalho, apresentando a descrição do problema que o *MonitorPrevIntel* propõe solucionar, sua motivação, seu objetivo, as justificativas para seu desenvolvimento e sua relevância. O Capítulo 2, “Tópicos Relevantes para o Sistema”, traz uma abordagem

⁷O nome *MonitorPrevIntel* foi uma adaptação sonora do termo extenso, que significa Monitor Inteligente de Prevenção.

sucinta sobre os conceitos e as tecnologias necessários à compreensão do objeto de estudo deste trabalho e suas futuras adaptações. No Capítulo 3, “Estado da Arte”, é apresentado o levantamento do estado da arte, explicitando as abordagens principais e mais interessantes encontradas para solução do problema de monitoramento remoto. Os aspectos metodológicos para construção do sistema são descritos no Capítulo 4, “Procedimentos Metodológicos”, bem como a modelagem e as tecnologias utilizadas para implementação do *MonitorPrevIntel*. O Capítulo 5, que recebe o nome do sistema inteligente, traz a descrição detalhada do sistema e suas partes, bem como, seu desenvolvimento e seu funcionamento. Os capítulos seguintes encerram o documento apresentando no Capítulo 6, “Testes e Resultados”, um caso de teste realizado para verificação do sistema e os resultados alcançados e no Capítulo 7, “Considerações Finais”, as considerações finais do trabalho, uma discussão em torno da comparação do sistema com os demais existentes no estado da arte e possibilidades futuras do *Monitor Intelligent Prevention*. Por último, as informações complementares ao sistema são apresentadas nos Apêndices e Anexos.

Capítulo

2

Tópicos Relevantes para o Sistema

“Toda a teoria só é boa na condição de que, utilizando-a, se vá mais além.”

André Gide

Neste capítulo é apresentado o contexto teórico das principais áreas temáticas que envolvem este trabalho: tecnologias utilizadas para transmissão sem fio, serviços de envio de mensagem através do celular (*Short Message Service* – SMS) e formas de representação do conhecimento.

2.1 Tecnologias para Transmissão Sem Fio

O uso de tecnologias móveis dissemina-se de maneira inimaginável. Mas afinal o que são estas tecnologias? MARKS *et al.* (2009) define os dispositivos portáteis, ou móveis, como sistemas computacionais que podem ser movidos fisicamente ou cujas capacidades podem ser executadas enquanto os mesmos estão em movimento. Estes dispositivos são conhecidos por trazer conforto e praticidade aos seus usuários, sendo atualmente utilizados em diversos equipamentos, celulares, palmtops, sensores, etc. Porém, mais que a mobilidade, o principal atrativo dos dispositivos móveis é a capacidade de conexão com redes *wireless*, fazendo com que seus usuários estejam conectados com qualquer pessoa no mundo, em qualquer lugar e a qualquer tempo.

A tecnologia de transmissão sem fio, ou *wireless*, é descrita por JINDAL, JINDAL e GUPTA (2005 apud MONSIGNORE, 2007 p.5) como “sistemas de telecomunicações em que ondas eletromagnéticas carregam sinais de dados sobre parte ou todo um trajeto de comunicação sem a utilização de meios físicos”. Nos últimos anos a velocidade com que se

desenvolveram as tecnologias de transmissão *wireless*, que começou de maneira lenta, tem dado passos largos. A cada ano as empresas de telecomunicações se aprimoram na criação de modelos que atendam as demandas das diversas áreas que passam a fazer uso desta tecnologia. Na área médica, as tecnologias de redes sem fio tornaram-se tendência por, através de seus inúmeros avanços, oferecer maior liberdade e possibilidades tanto para pacientes quanto para profissionais da saúde.

A escolha de determinada tecnologia *wireless* deve considerar algumas características básicas, tais como: alcance, velocidade, capacidade de transmissão, quantidade máxima de nós que podem ser distribuídos em uma rede, tempo de vida da bateria e tamanho dos dispositivos. No contexto do monitoramento de sinais vitais humanos, a escolha do tipo de transmissão é definida de acordo com as características do sistema a ser implementado. De maneira geral, as características observadas são: disponibilidade, segurança dos dados, rápida entrega dos dados diante de uma situação de emergência, robustez e mobilidade (TEIXEIRA, 2009).

Alguns dos principais protocolos e especificações técnicas para uso em redes sem fio são Wi-Fi (IEEE 802.11), Z-Wave, Bluetooth e ZigBee. Porém, devido a alta largura de banda, o maior consumo de energia e o alto preço do protocolo Wi-Fi e a especificidade da tecnologia proprietária Z-Wave para automação residencial, as tecnologias mais adequadas para uso em redes para monitoramento de sinais vitais são ZigBee e Bluetooth. Estas últimas tecnologias têm por base o padrão IEEE 802.15 que se caracteriza por um conjunto de padrões para Redes Pessoais Sem Fio (*Wireless Personal Area Network* – WPAN). As redes deste tipo foram criadas da necessidade de substituir cabos de curta distância (SANTOS, 2007), geralmente até 10 metros, por transmissões *wireless* de baixa transmissão de dados, baixo consumo e funcionamento de pequena complexidade (MARÇAL, 2008). Os sistemas de monitoramento humano, em sua maioria, fazem uso de alguma implementação de WPAN.

2.1.1 Tecnologia ZigBee

O padrão IEEE 802.15.4 foi lançado em 2003 para suprir as necessidades emergentes das WPAN. Em 2004, um grupo de empresas de diferentes segmentos de mercado, chamado *ZigBee Alliance*, formou uma aliança para promover este padrão (CASTRO *et al.*, 2010). Este protocolo tenta atender as especificações das aplicações de monitoração, sensoriamento e

controle de sistemas das mais diversas áreas, tais como: automação residencial, comercial e industrial, controle de diferentes dispositivos eletrônicos e monitoração de ambientes e de pacientes.

Projetado para interconexão de pequenos dispositivos de comunicação de baixo custo e para controle e segurança de redes de sensores sem fio de baixa complexidade e potência, esta tecnologia traz aos sistemas flexibilidade, mobilidade e facilidade no uso (ZIGBEE, 2011). A tecnologia ZigBee é um padrão para dispositivos de baixo consumo de energia, pequena latência, baixa velocidade de transmissão, com redes auto reconfiguráveis, de curto alcance, com suporte a milhares de dispositivos. As frequências de onda usadas para transmissão de dados no padrão ZigBee são faixas de frequência livre, isto é, não requerem licença para uso. Em geral, usam-se as bandas de rádio industriais, científicas e médicas (*Industry, Scientific and Medical – ISM*) (PINHEIRO, 2006).

O protocolo ZigBee é mantido e desenvolvido, em conjunto, pela ZigBee Alliance, que atualmente conta com mais de 200 empresas, e pelo IEEE. De acordo com o modelo de referência OSI, este protocolo está dividido em cinco camadas (Figura 2.1): camada física (*Physical – PHY*) e camada de controle de acesso (*Medium Access Control – MAC*), definidas pelo padrão IEEE 802.15.4, camada de rede e segurança (*Network – NWK*) e camada de suporte a aplicação, definidas pela ZigBee Alliance, e a camada de aplicação, definida pelo usuário (PINHEIRO, 2006).

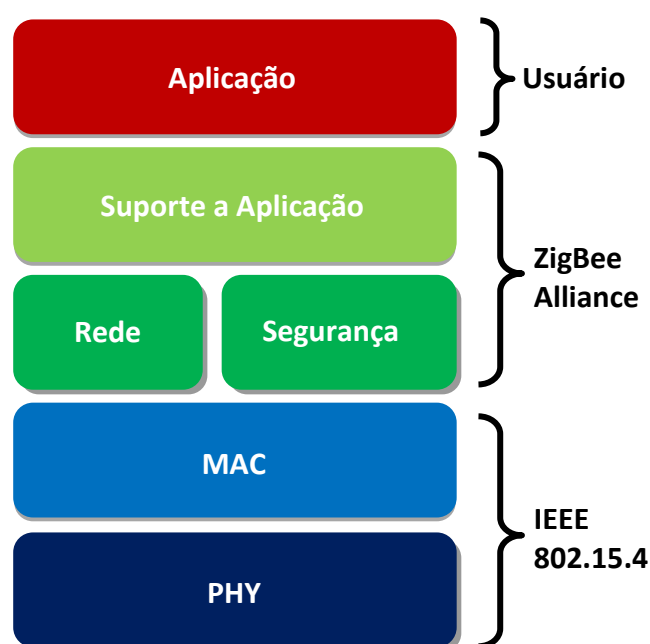


Figura 2.1 - Pilha de protocolos ZigBee.

Segundo GASCÓN (2008), é possível dividir os dispositivos ZigBee de uma rede em três tipos: coordenador (*coordinator*), roteadores (*routers*) e nós finais (*end devices*). O coordenador é o dispositivo mestre na rede, aquele que a controla. Os roteadores fazem o roteamento das informações enviadas por um nó final. Os nós finais são sensores que captam a informação do ambiente. Note que em uma rede tem-se no máximo um dispositivo coordenador, para centralizar o ponto de controle, e que os nós configurados como nós finais não devem se comunicar entre si.

Ainda relacionado ao Zigbee, o padrão suporta três tipos de topologia de rede (Figura 2.2): estrela, árvore e malha (ZIGBEE, 2008). Na primeira, a rede é composta por um coordenador e por nó(s) final(-is). O coordenador inicializa e mantém os demais dispositivos que estão diretamente conectados a ele. Nas duas últimas topologias temos os três tipos de dispositivos já citados. O coordenador inicializa a rede e escolhe certos parâmetros de configuração, mas a rede tem seu alcance estendido com a utilização de roteadores como nós centrais de redes menores de nós finais. A diferença entre a topologia do tipo árvore e a do tipo malha é que a última é capaz de se ajustar automaticamente quando da entrada ou saída de um novo dispositivo na rede e a possibilidade de implementação de diferentes caminhos entre dois nós.

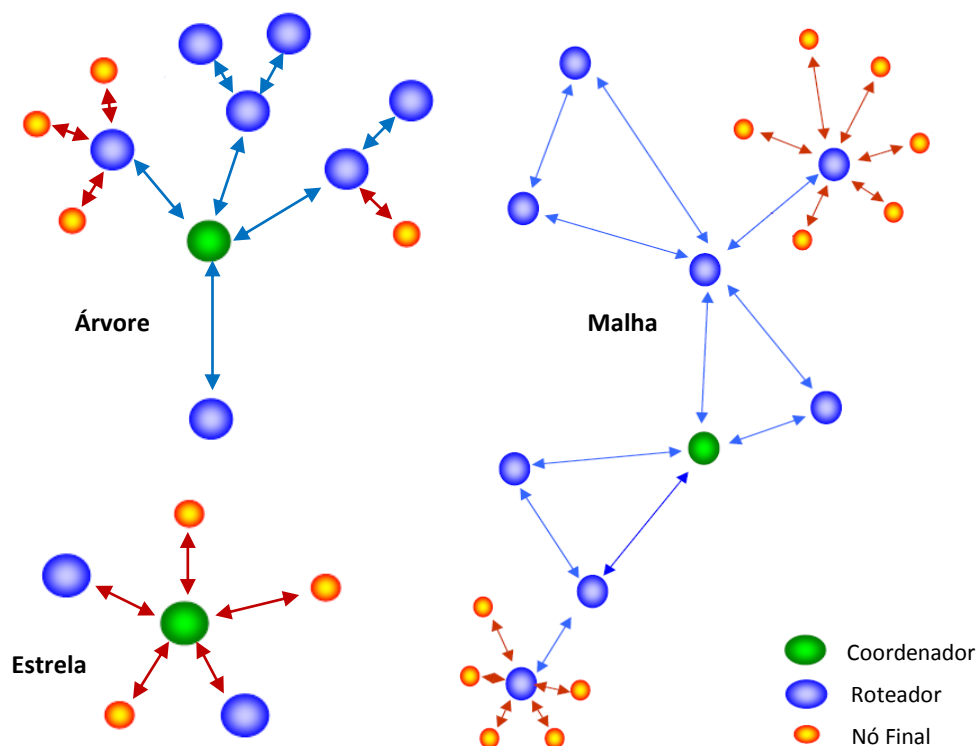


Figura 2.2 - Topologias das redes ZigBee.

Além disso, a disposição dos dispositivos ZigBee na rede do tipo malha ainda permite que uma rede se configure automaticamente quando da entrada ou retirada de um nó. Mas apesar de todas as vantagens apresentadas, esta tecnologia não é utilizada por nenhum telefone móvel. Então não é possível criar uma associação de sensores que utilizam esta tecnologia com o dispositivo para o qual é desenvolvida a aplicação deste trabalho.

2.1.2 Tecnologia Bluetooth

No ano de 1994, a empresa sueca Ericsson começou a estudar tecnologias sem fio acessíveis que permitissem a substituição dos cabos utilizados para conectar os telefones celulares e seus acessórios (STEIN, 2003). Segundo Alecrim (2011), estes estudos que tomaram por base os mecanismos de comunicação nas redes de telefonia móvel resultaram num projeto com o nome de *MCLink* que utilizava sinais de rádio de baixo custo e curto alcance.

Em meados de 1997, outras empresas começaram a se interessar pelo projeto da Ericsson e em maio de 1998, essa empresa se uniu à IBM, Intel, Toshiba e a Nokia. Tais empresas anunciaram formalmente a formação do *Bluetooth Special Interest Group* (SIG), um consórcio criado especialmente para cuidar da especificação e desenvolvimento do que seria o padrão Bluetooth. A junção dessas gigantes do ramo de telecomunicações, do desenvolvimento de computadores e de chips processadores possibilitou o desenvolvimento de padrões que garantem a interoperabilidade das tecnologias dos mais diversos dispositivos (ALECRIM, 2011). De acordo com o Bluetooth SIG, o consórcio atualmente conta com mais de 14000 empresas de áreas como telecomunicações, computação, redes e eletrônicos.

Projetado para ser simples, robusto, de baixo consumo e baixo custo (BLUETOOTH SIG, 2011a), Bluetooth é um padrão de tecnologia wireless para comunicação de maneira simples, segura e abrangente, de diversos dispositivos nas WPAN. Um dos poucos padrões no qual são definidas desde as camadas mais baixas da rede até a camada de aplicação da pilha de protocolos (Figura 2.3). Além disso, a tecnologia Bluetooth implementa diferentes mecanismos de segurança para trazer confidencialidade, disponibilidade e integridade, tais como: autenticação, protocolos de chave de segurança, como o *Elliptic Curve Diffie Hellman* (ECDH) e algoritmos de cifra de blocos SAFER+.

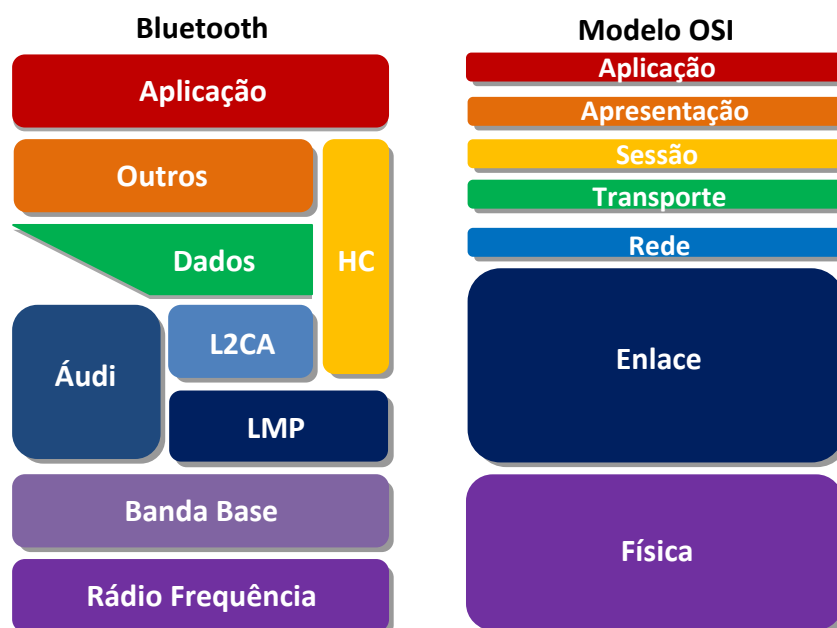


Figura 2.3 - Pilha de protocolos do Bluetooth x Modelo OSI.

Para transmissão de informações é utilizada uma tecnologia de rádio baseada em saltos de frequência adaptativa (AHF, do inglês *Adaptive Hopping Frequency*) com espectro espelhado. Com o uso desta tecnologia de rádio, o meio de transmissão é dividido em vários canais, por onde os dados são transmitidos usando diferentes bandas de frequência para tentar reduzir a interferência com outras transmissões que usem a mesma faixa de frequência (BLUETOOTH SIG, 2011b), já que as ondas usadas pelo Bluetooth são as ISM, livres de licença e usadas por diferentes tecnologias. Em geral, a banda usada ocupa o intervalo de 2.4Ghz a 2.485Ghz dividido em 79 canais com 1MHz de intervalo entre eles, com um sinal *full-duplex* (transmite e recebe dados) usando um esquema de *Time Division Duplex* (TDD). Em países onde a faixa ISM é menor, o Bluetooth usa um número menor de canais (23).

Os dispositivos Bluetooth são categorizados de acordo com o alcance máximo em três classes. A Classe 1 tem potência máxima de 100mW e alcance de até 100 metros. A Classe 2 tem potência de até 2,5mW e alcance máximo de 10 metros. A última classe, a Classe 3, usa potência máxima de 1mW e alcance de até 1 metro. Com estas categorias, dois dispositivos podem se comunicar através de diferentes distâncias observadas as restrições de suas classes. Por exemplo, dois dispositivos de Classe 3 só podem se comunicar se tiverem uma distância máxima de 1 metro entre eles.

Para iniciar a comunicação, um dispositivo Bluetooth, que será chamado de servidor, anuncia quais os serviços suportados ou oferecidos por ele aos demais dispositivos

encontrados na sua área de alcance, que serão potenciais clientes⁸. Em seguida um dos clientes encontrados responde ao servidor qual serviço irá utilizar. Para um usuário que utilize um dispositivo com Bluetooth, como um aparelho celular, por sua vez, não precisa configurar qualquer endereço de rede e, em alguns casos, nem mesmo precisa permitir a comunicação dos dispositivos. Esta facilitação da descoberta de dispositivos e de sua configuração simplifica de maneira considerável o uso da tecnologia.

Esta comunicação entre dispositivos que utilizam a tecnologia Bluetooth é chamada de *pairing*, ou pareamento, pois, para que ela ocorra, um dispositivo envia uma requisição para outro dispositivo que responde a requisição e ambos criam um canal de compartilhamento secreto através de chave pública, chamado de *link key*. Segundo (BLUETOOTH SIG, 2010), dois ou mais dispositivos Bluetooth que compartilham o mesmo canal formam uma rede chamada *piconet*. Nessas redes um dispositivo age como *master* e os demais dispositivos atuam como *slave*, sendo que em cada *piconet* pode haver até sete *slaves* ativos. Cada *slave* pode fazer parte de uma ou mais *piconets*, mas cada *master* só pode atuar como tal em uma *piconet*. Quando várias redes destas existem numa mesma área com conexões entre elas, dizemos que há uma *scatternet* (Figura 2.4).

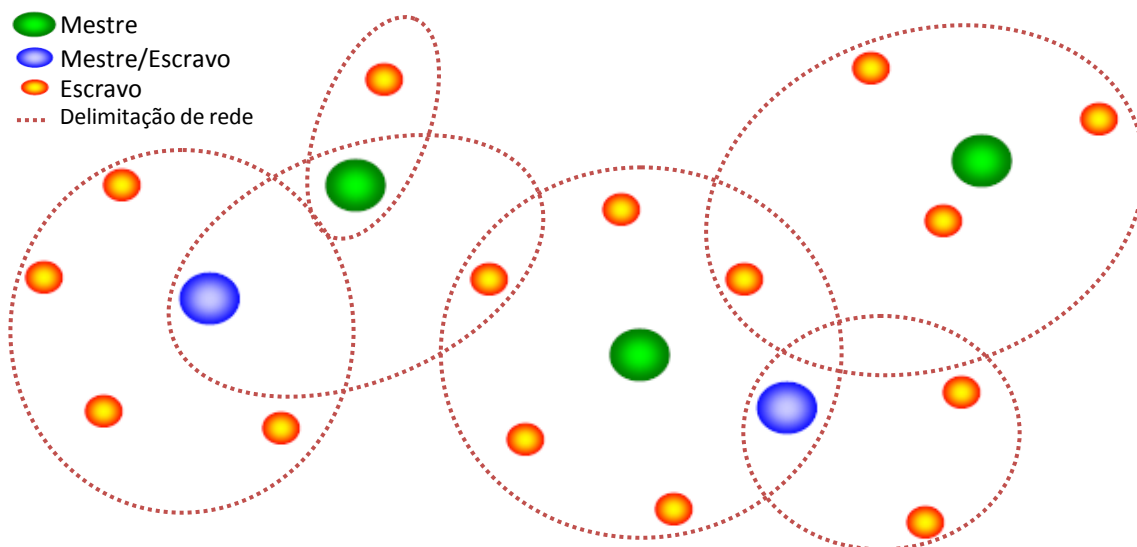


Figura 2.4 - Scatternet Bluetooth com seis redes.

Uma característica bem interessante desta tecnologia é que ela está acessível e permite a comunicação entre os mais diferentes dispositivos, desde telefones celulares, fones de

⁸ Cliente é um sistema computacional que consome, utiliza serviços de um servidor. Um cliente potencial são aqueles que conseguem se comunicar com o servidor, mas não necessariamente irá utilizar o (s) serviço (s) disponibilizados por esse.

ouvido, até computadores, impressoras e dispositivos na área médica. Bluetooth é uma tecnologia que evolui constantemente com surgimento de novas versões e diferentes especificações, mas que procura preservar a compatibilidade entre os padrões. A última versão lançada, a 4.0, une a tecnologia da versão 3.0 HS (*High Speed*) com uma nova tecnologia que tem foco no baixo consumo de energia passando a dividir os sistemas de tecnologia Bluetooth em dois grupos: *Basic Rate* (também chamado de *Classic Bluetooth*) e *Low Energy* (BLUETOOTH SIG, 2010).

2.1.2.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE), um subconjunto da especificação do Bluetooth 4.0, foi desenvolvido para atender as necessidades dos dispositivos de baixa energia e baixa potência. Com isso, este novo padrão consegue abranger uma maior diversidade de aparelhos: relógios, controles remotos, equipamentos médicos e de saúde, fones de ouvido, *joysticks*, telefones celulares, *smartphones*, computadores, etc. Fazendo aumentar a cada dia a convergência de diferentes tecnologias para este protocolo de comunicação.

A versão BLE otimizou o protocolo para atender aos equipamentos de menor consumo de corrente, menor complexidade, menor custo, menores taxas de dados e menores taxas de ciclos. Segundo o diretor executivo do SIG, Mike Foley, o *Bluetooth Low Energy* é indicado para operar sobre equipamentos que requerem o máximo do tempo de vida da bateria. Os novos chips permitem que alguns dos dispositivos de baixa energia durem por muitos anos usando uma pequena célula de bateria, do tamanho de um botão (MELLO JÚNIOR, 2010). Estes chips conseguem obter um consumo de 0,5 a 0,01 da energia de um chip do *Classic Bluetooth* (BLUETOOTH SIG, 2011b).

Para ficar mais acessível aos equipamentos, esta tecnologia refinou vários aspectos de seus antecessores. Uma destas características é o uso de dois esquemas de múltiplo acesso, *Frequency Division Multiple Access* (FDMA) e *Time Division Multiple Access* (TDMA). O primeiro esquema divide a banda de frequência ISM, usando a mesma faixa já usada pelo *Classic Bluetooth* (de 2.4Ghz a 2.485GHz), em 40 canais com intervalos de 2MHz. E o TDMA toma por base um esquema de captação onde um dispositivo transmite um pacote em um tempo pré-determinado e o dispositivo que recebe o pacote responde após um intervalo de tempo, também pré-determinado.

A comunicação nos dispositivos do tipo BLE também difere da usada nos do tipo *Classic Bluetooth*. Enquanto este último utiliza um código de acesso, chamado de *link key*, o outro utiliza o endereço de acesso (*Access Address*) do dispositivo alvo, que é único para cada dispositivo, para fazer o *pairing*. Isso faz com que vários dispositivos possam compartilhar um mesmo canal físico sem interferir na comunicação com os demais. A especificação BLE diz que um dispositivo só é capaz de estar conectado a um único canal físico por vez, mas admite que dispositivos mais avançados possam estar conectados simultaneamente a mais de um canal físico (BLUETOOTH SIG, 2010).

Além disso, esta nova tecnologia tem seu alcance ampliado para até 200 metros, diminui o tempo de latência de mudança para o estado conectado, utiliza criptografia AES com *Counter Mode CBC-MAC* de 128 bits, entre outros aspectos. Na Tabela 2.1 podemos ver um comparativo entre o *Bluetooth Low Energy*, o *Classic Bluetooth* e o *ZigBee*.

Tabela 2.1 - Comparativo entre Classic Bluetooth, Bluetooth Low Energy e ZigBee.

	Classic Bluetooth	Bluetooth Low Energy	ZigBee
Normalização	Bluetooth SIG	Bluetooth SIG	ZigBee Alliance
Padrão de Rede	IEEE 802.15.1	IEEE 802.15.1	IEEE 802.15.4
Alcance Máximo	1 - 100 m	200 m	150 m
Frequência de Operação	2,4 – 2,485Ghz	2,4 – 2,485 Ghz	2,4 Ghz
Taxa de Dados	1 - 3 Mb/s	1 Mb/s	250 Kb/s
Consumo de Potência	0 - 100 mW	0,01 mW	1 mW
Quantidade de Nós por Rede	8 - (dos quais, 7 escravos podem estar ativos por vez)	8 - (dos quais, 7 escravos podem estar ativos por vez)	64000

Apesar das características apresentadas, ainda não é viável o uso do BLE devido a sua incompatibilidade com as versões anteriores do Bluetooth. Por isso, no mercado passarão a ser vendidos três tipos de chips Bluetooth: os que implementam os dois sistemas, os que implementam o BLE e os que implementam o *Classic Bluetooth*. Assim, equipamentos que disponham do chip dual poderão se comunicar com chips BLE e chips clássicos, mas chips que implementam um único sistema só poderão se comunicar com chips do seu tipo (BLUETOOTH SIG, 2010). Quando for iniciada a fabricação desse chips BLE será possível utilizar a comunicação Bluetooth para transmissão das medidas de sinais vitais dos sensores

para o aparelho celular, viabilizando o uso desse para aplicações médicas como o *MonitorPrevIntel*.

2.2 Short Message Service (SMS)

SMS é um serviço de envio e recebimento de mensagens curtas (AYABE; CHANDER; MIZIKOVSKY, 2000) através de redes de telecomunicações, que tem sua origem na rádio telegrafia e nos *paggers*. Criado como parte do padrão GSM (*Global System for Mobile Communications*) desde 1985 pela ETSI (*European Telecommunications Standards Institute*), o serviço só passou a ser utilizado na década de 1990 quando em 1992, no Reino Unido, Neil Papworth do Sema Group enviou a mensagem *Happy Christmas* para o *handset* Orbitel 901 de Richard Jarvis da empresa Vodafone usando um computador (HPPY BTHDY TXT, 2002). Atualmente o 3GPP (*Third Generation Partnership Project*) é responsável por desenvolver e manter esta tecnologia.

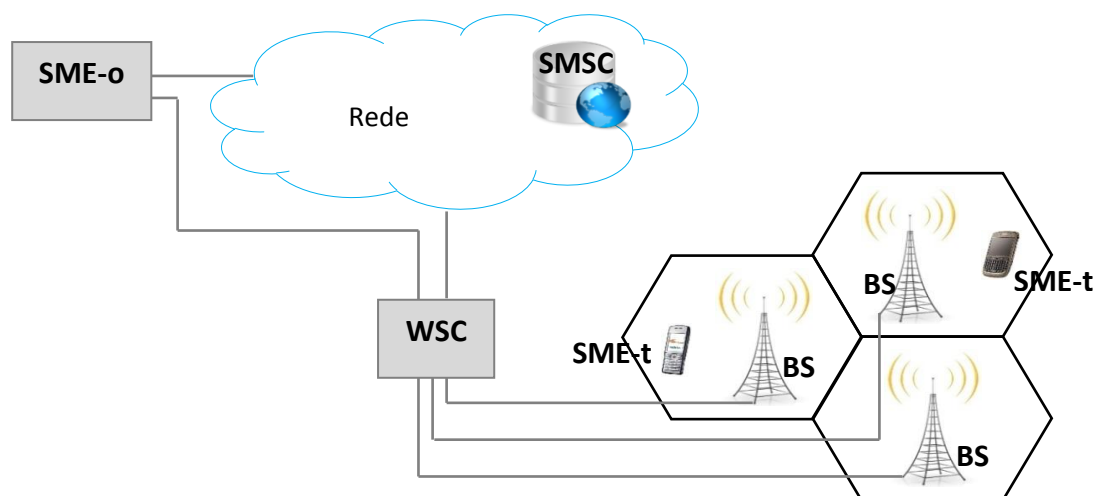


Figura 2.5 - Arquitetura SMS.

A arquitetura desse serviço (Figura 2.5) pode incluir até quatro tipos de dispositivos: as entidades de mensagens curtas (*Short Message Entities*, SME), centros de comutação sem fio (*Wireless Switching Center*, WSC), estações base (*Base Stations*, BS) e os centros de SMS (*SMS Center*, SMSC). As entidades de mensagem curta, que são dispositivos capazes de compor, exibir ou eliminar mensagens, podem ser entidades originadoras (telefones móveis, *paggers* ou computadores) ou terminadoras (telefones celulares ou *paggers*) (MOKHTAIAN,

2007). Os centros de comutação sem fio recebem a mensagem através de uma rede ou diretamente de uma SME e a roteia através de uma *Base Station* até uma SME terminadora (SME-t).

Cada estação base é uma antena comunicável com os WSC responsável por encaminhar as mensagens desses até as entidades terminadoras de sua região (cada região é chamada de célula). O SMSC é usado quando uma SME originadora (SME-o) envia sua mensagem de forma indireta até o WSC ou quando a SME-t não está pronta para recebimento da mensagem. Assim, o centro SMS é um servidor *store-and-forward* que recebe as mensagens, as armazena e as encaminha até que a SME-t esteja pronta para recebê-las.

Como a *Telecommunications Industry Association* (TIA), associação comercial representante das indústrias globais de Tecnologia da Informação e Comunicação (TIC), recomenda padrões provisórios para diferentes interfaces e redes sem fio (AYABE; CHANDER; MIZIKOVSKY, 2000), não existe um protocolo único para envio de dados através das redes. Assim diversos formatos de pacotes trafegam pelas redes de telecomunicações. Além disso, o texto das mensagens SMS pode ser codificado utilizando padrões alfabéticos diferentes. Os mais comuns são o de 7-bits GSM, o de dados de 8-bits e o de 16-bits UTF-16. De acordo com o padrão utilizado pelo aparelho celular, será calculado o tamanho máximo de uma mensagem SMS enviada por ele. Para os padrões citados as mensagens serão de 160, 140 ou 70 caracteres, respectivamente.

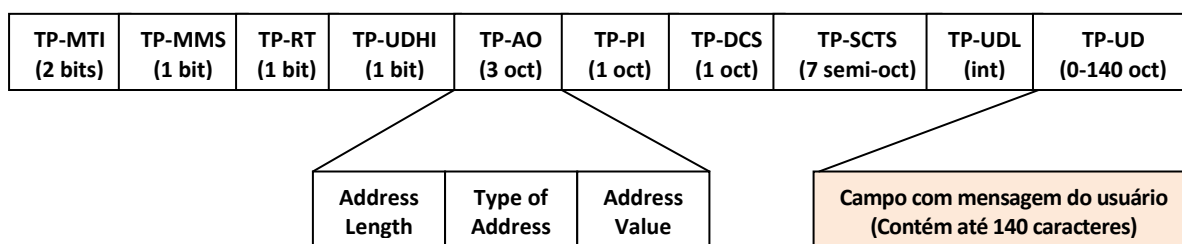


Figura 2.6 - Esquema de um Protocol Data Unit.

Para definir como o SMS deve operar, o ETSI estabeleceu vários documentos para descrever como deve funcionar a tecnologia SMS nos celulares. Entre eles, o principal é o documento GSM 03.40 que estabelece os serviços e seus elementos, a arquitetura da rede, as funcionalidades do SMSC e do MSC, os requisitos de roteamento, os protocolos e suas camadas (ETSI, 1996b). Também nesse documento é especificado o formato das mensagens SMS para envio por GSM, o *Protocol Data Unit* (PDU), que pode ser visto na Figura 2.6. A estrutura básica de uma mensagem no modo PDU contendo dados de usuário, chamado de

Transfer Protocol Data Unit (TPDU) ou ainda de *SMS-Deliver*, divide o pacote em dez campos:

- *TP-Message-Type-Indicator* (TP-MTI): 2 bits que indicam o tipo da mensagem sendo enviada, o *SMS-Deliver* será representado por 00;
- *TP-More-Messages-to-Send* (TP-MMS): bit que indica se há mais mensagens para envio (1) ou não (0);
- *TP-Reply-Path* (TP-RP): bit que indica se existe caminho de resposta (1) ou não (0);
- *TP-User-Data-Header-Indicator* (TP-UDHI): bit que indica se o campo TP-UD tem (1) ou não (0) cabeçalho;
- *TP-Originating-Address* (TP-OA): endereço da SME originadora da mensagem que é composta por três subcampos: *Address-Length*, *Type-of-Address* e *Address-Value* que indicam respectivamente o tamanho do número, o tipo do número – se é no formato internacional (indicado por 91 hexadecimal) ou nacional (indicado por 81 hexadecimal), por exemplo, e o número propriamente dito da SME;
- *TP-Protocol-Identifier* (TP-PI): octeto usado para identificar o protocolo da camada acima, caso exista;
- *TP-Data-Coding-Scheme* (TP-DCS): octeto usado para identificar o esquema de codificação do TP-UD, pelo GSM 03.38, o ETSI estabeleceu o alfabeto de 7-bits GSM como alfabeto obrigatório em telefones móveis e em dispositivos de rede de telecomunicações (ETSI, 1996a);
- *TP-Service-Centre-Time-Stamp* (TP-SCTS): parâmetro composto por sete semi-octetos (cada um composto por dois dígitos decimais) (PETTERSSON, 2005) que indicam, respectivamente, ano, mês, dia, hora, minuto, segundo e zona do tempo (de acordo com o GMT) do instante do recebimento da mensagem pelo centro de serviço;
- *TP-User-Data-Length* (TP-UDL): número inteiro que indica o tamanho do campo TP-UD;
- *TP-User-Data* (TP-UD): campo que contém a mensagem que o usuário compôs, podendo conter até 140 octetos (usando o alfabeto de 7-bits, tem-se 160 posições).

Para o usuário, o uso de pequenas mensagens, através de interfaces ubíquas provendo comunicação instantânea e ainda permitindo um mecanismo de *store-and-forward* quando não for possível o envio imediato, faz desta tecnologia um dos serviços de comunicação mais bem

sucedidos no mundo (MOKHTAIAN, 2007; ZERFOS et al, 2006). Para outros, que utilizam o SMS como fornecedor de serviço o uso é justificado pelo preço. A média mundial para uma mensagem é US\$ 0.11. Caso o envio seja entre redes telefônicas diferentes, este custo ainda recebe uma taxação de £ 0.03 (SHAH *et al.*, 2010).

A grande crítica em relação a esse serviço é a ausência de segurança. Sobre isso várias abordagens são propostas, tais como criar um canal seguro de comunicação (CRUZ; PEREIRA; SILVA, 2008) ou criptografar o conteúdo da mensagem (MARDIKAR, 2009; LO; LUO; YEOH, 2009; AHMEDA; EDWILA, 2009). Porém, cabe analisar o cenário da aplicação para determinar qual a abordagem mais adequada.

2.3 Inteligência Artificial

Inteligência Artificial (IA) é um ramo da Ciência da Computação que tenta representar o conhecimento e os métodos de raciocínio sobre o mesmo no computador. Existem diversas definições, com diferentes pontos de vista, para o termo IA. Russel e Norvig (2003) reúnem alguns conceitos categorizando-os em quatro grupos: sistemas que pensam como humanos, que agem como humanos, que pensam racionalmente e que agem racionalmente (Figura 2.7).

Sistemas que pensam como Humanos	Sistemas que pensam racionalmente
“O novo e excitante esforço para tornar os computadores pensantes... <i>máquinas com mentes</i>, no sentido completo e literal.” (Haugeland, 1985) “[A automação de] atividades que nós associamos a pensamentos humanos, atividades como tomada de decisões, resolução de problemas, aprendizado...” (Bellman, 1978)	“O estudo de faculdades mentais através do uso de modelos computacionais.” (Chamiak e McDermott, 1985) “O estudo de computações que tornam possível percepção, razão e ação.” (Winston, 1992)
Sistemas que agem como Humanos	Sistemas que agem racionalmente
“A arte de criar máquinas que desempenham funções que requeiram inteligência quando executadas por pessoas.” (Kurzweil, 1990) “O estudo de como fazer computadores fazerem coisas que, no momento, as pessoas são melhores.” (Rich e Knight, 1991)	“Inteligência Computacional é o estudo do design de agentes inteligentes.” (Poole <i>et al.</i>, 1998) “IA... preocupa-se com comportamento inteligente em artefatos.” (Nilson, 1998)

Figura 2.7 - Algumas definições para IA, agrupadas em quatro categorias (RUSSEL; NORVIG, 2003, p. 2, tradução nossa)

Inúmeras linguagens, modelos e mecanismos de inteligência foram criados para auxiliar a construção de componentes computacionais inteligentes, alavancados pelos imensos avanços tecnológicos. Tais avanços permitiram que a IA fosse utilizada para diversos fins, desde programas de computador, aplicativos de segurança computacional, jogos, desenvolvimento da robótica, dispositivos para reconhecimento de face, fala e escrita, até programas de auxílio a diagnósticos médicos.

Para implementar técnicas de inteligência artificial geralmente são utilizados *softwares* de comportamento autônomo, denominados Agentes Inteligentes (AI). Um agente é uma entidade computacional que analisa e modela sistemas, programa e resolve problemas, funcionando de forma contínua e autônoma em certo ambiente. Segundo PALAZZO (2005), agentes são capazes de ação autônoma flexível, incluindo reatividade, aprendizado, proatividade e habilidade social, bem como armazenar e manipular dados, adquirir, representar e manipular conhecimentos (CÂMARA, 2000 ou 2001). O agente atua com o propósito de realizar ações que são entendidas como benéficas ao seu objetivo final.

Um AI é qualquer sistema que percebe informações do ambiente no qual está inserido através de sensores e age por meio de estruturas, chamadas atuadores (RUSSEL; NORVIG, 2003, p.32), na Figura 2.8 podemos ver uma de suas possíveis arquiteturas. Alguns tipos de agentes inteligentes também maximizam o desempenho de processamento, isto é, o grau de sucesso de atuação sobre um dado ambiente. Para isso, eles agem formulando objetivos baseados na sua situação atual (seu estado) e na medida de desempenho para determinar seus comportamentos.

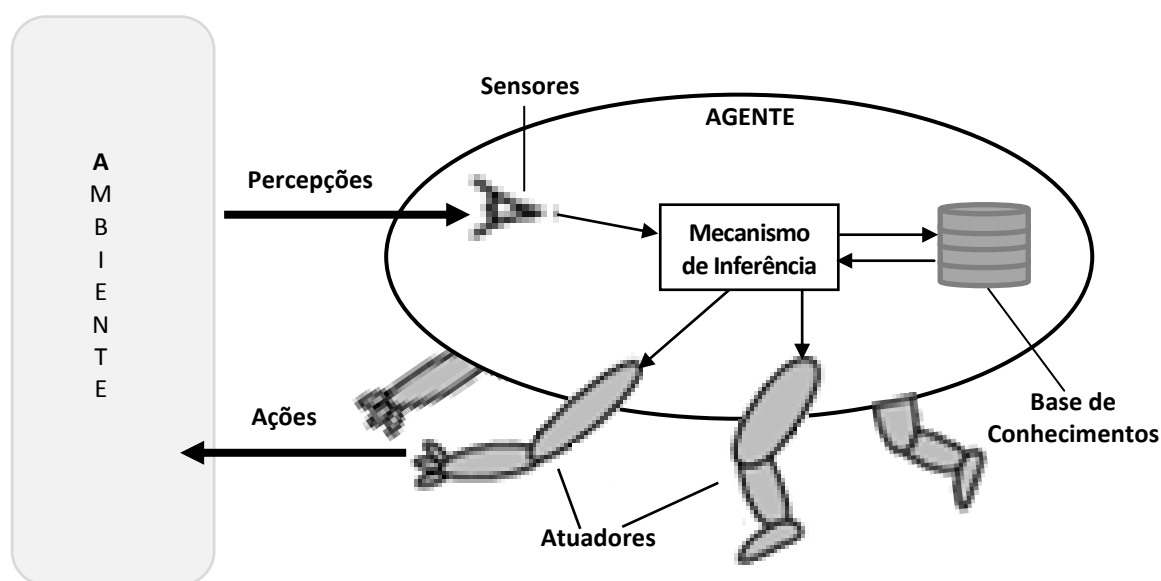


Figura 2.8 - Arquitetura de agente reflexivo em interação com o ambiente.

Para construção de um AI é necessário considerar diversos aspectos, tais como: qual o foco do sistema dentre as quatro grandes áreas apontada na Figura 2.7, a escolha do mecanismo de raciocínio e da arquitetura a ser utilizada, como se dará a interação com o ambiente, como será representado o conhecimento deste agente. Este último aspecto é uma das principais questões relacionada à representação do conhecimento, isto é como armazenar e transmitir, de maneira adequada, o conhecimento para um sistema computacional. Em outras palavras, como sistematizar o conhecimento em fórmulas lógicas que consigam exprimir todas as características e exceções relativas ao domínio do problema ao qual o agente em questão pretende solucionar. Câmara (2000 ou 2001) apresenta algumas das principais questões.

De que forma o conhecimento pode ser expresso?
Como encontrar a linguagem adequada para a representação deste conhecimento?
Como formar uma base de conhecimento suficientemente detalhada e que represente a compreensão do domínio?
Como realizar inferências automáticas, dando acesso tanto ao conhecimento implícito na base de conhecimento quanto àquele armazenado explicitamente (declarativo)?
Como o sistema deve proceder na presença de informações incompletas, incorretas ou de senso comum? (CÂMARA, 2000 ou 2001)

Diante de tantos questionamentos, em torno de como implantar IA em sistemas computacionais, foi possível perceber que os mecanismos de inteligência e compreensão do mundo pelos seres humanos são muito mais complexos do que poderiam imaginar os precursores de seu estudo. Bem como, seria impossível conseguir modelar o pensamento humano por completo em componentes computacionais. Foi então que os pesquisadores começaram a desenvolver sistemas mais realistas, substituindo o foco de seus estudos para a compreensão de atividades mais práticas dando origem aos Sistemas Especialistas (SE) (NAVEGA, 2000). Exemplos de sistemas especialistas são os sistemas de reconhecimento de padrões, mineração de dados, entre outros.

O objetivo de um Sistema Especialista é utilizar os conhecimentos de um especialista humano para executar alguma atividade específica desse. Em outras palavras, o SE armazena e sequencia a informação obtida de um ou mais especialistas humanos para inferir sobre problemas específicos a estes especialistas e tenta solucioná-los. Para construir esse sistema eficientemente é necessária a representação correta e não ambígua do conhecimento no sistema. Para isto, segundo LINDER (2011), são utilizados passos de coleta (ou aquisição), modelagem (ou formalização) e implementação (ou codificação).

Esses são os passos da representação do conhecimento. A coleta envolve a aquisição e acúmulo da fonte do conhecimento, que pode ser especialistas, livros, documentos, filmes,

etc. Esta tarefa de aquisição costuma ser bastante complexa, pois nem sempre o conhecimento está explícito, verbalizável, dedutível ou completo, dificultando a construção de um conjunto consistente de dados. Seguida a coleta vem a modelagem do conhecimento acumulado, na qual cria-se um modelo formal para representar os dados do especialista no computador. Por último, a partir do modelo criado anteriormente, os dados são codificados e armazenados no sistema computacional. Ao conjunto de dados coletados da fonte especialista e armazenados no computador é dado o nome de base de conhecimentos.

Os SE utilizam essa base de conhecimentos para solucionar problemas simulando o comportamento do especialista humano, a partir de aprendizado, análise, controle, interpretação, aconselhamento, monitoração, comunicação, classificação, diagnóstico, testes, entre outras ações. A estrutura de um sistema especialista se assemelha ao modelo de Sistema Simbólico de NEWELL (1980) e pode ser dividida basicamente em: memória de trabalho, base de conhecimento e mecanismo de inferência. Na Figura 2.9 é apresentada uma comparação entre o modelo de Newell e um modelo genérico de um sistema especialista.

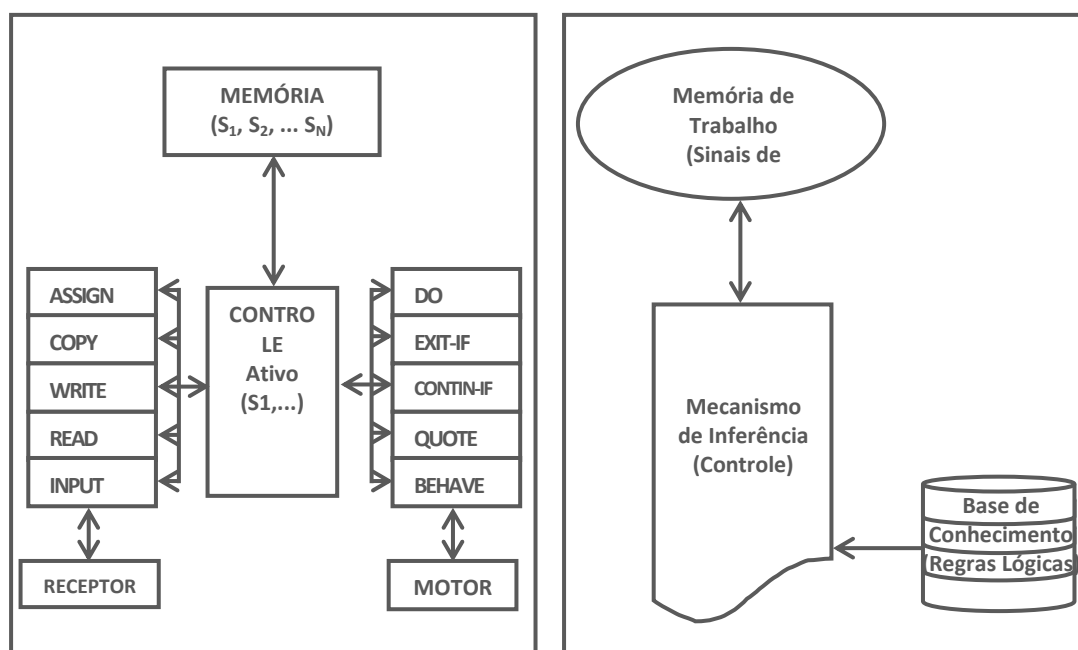


Figura 2.9 - Comparação modelo Newell (NEWELL, 1980, p. 143, tradução nossa), à direita, e modelo de SE, à esquerda.

A memória de trabalho é uma memória volátil, pois registra os dados de um problema a ser resolvido em um dado momento, isto é, um problema em particular. Ela guarda os dados que serão interpretados pelo SE. A base de conhecimentos armazena as informações coletadas dos especialistas sobre uma estrutura de fatos, regras e heurísticas (LINDER, 2011). Essas

serão as informações utilizadas para interpretar os dados da memória de trabalho e solucionar o problema. O conhecimento está armazenado utilizando alguma linguagem escolhida através do modelo de formalização utilizado para a representação computacional do conhecimento. Já o mecanismo de inferência é uma estrutura de controle da atividade do sistema. Esse mecanismo é responsável por buscar a correspondência dos dados da memória de trabalho com os fatos e as regras da base de conhecimentos, solucionar possíveis conflitos entre as regras selecionadas e executar as ações do SE com base nas regras.

Existem diversas abordagens para escolha da representação do conhecimento na base e para a resolução de conflitos, desde selecionar a mais utilizada até executar todas segundo a ordem de prioridade. Uma técnica bastante utilizada para a representação, que se baseia na linha lógica, são as regras de produção. Uma regra de produção é uma estrutura lógica que se divide em uma pré-condição sensitiva e uma ação realizada de acordo com o valor verdade da pré-condição. Ou ainda, como colocam RUSSELL e NORVIG (2003), as regras se baseiam em dois axiomas, chamados de *possibility axiom* e *effect axiom*. O primeiro axioma define uma condição a ser verificada pelo motor de inferência com a memória de trabalho e quando atendida deve desencadear o *effect axiom* ação correspondente à regra de produção ou as mudanças a serem executadas.

A estrutura lógica utilizada pelas regras de produção para descrever um conhecimento é do tipo “SE <pré-condição(ões)> ENTÃO <ação(ões)>”. As pré-condições são os *possibility axiom* e as ações são os *effect axiom*. O uso de regras de produção justifica-se pela sua semelhança com o processo de inferência humana, a sua facilidade de implementação, a modularidade e a possibilidade de atualização da base de conhecimentos. Um exemplo de regra de produção derivada de conhecimento humano pode ser vista na Tabela 2.2.

Tabela 2.2 - Exemplo de codificação de conhecimento em regra de produção.

Conhecimento	Regra de Produção
Todo homem é mortal.	SE <i>entrada</i> É <i>homem</i> ENTÃO <i>entrada</i> É <i>mortal</i>

Além disso, existem algumas ferramentas para desenvolvimento de aplicações que utilizam base de regras de produção. Entre as quais, para este trabalho, merece destaque o KEOPS, que é um sistema de regras de produção voltado para uso em telefones móveis que toma por base a linguagem Java. Seu nome vem do acrônimo para *Embedded Objects Production Systems* e o K inicial remete a máquina virtual Java para aparelhos celulares. Segundo seu criador ALBUQUERQUE (2002), esta ferramenta é uma extensão da arquitetura

de desenvolvimento de sistemas de produção para a linguagem Java, o JEOPS (*Java Embedded Objects Production Systems*). Esse sistema permite a escrita de regras de produção em uma linguagem muito próxima à linguagem Java, onde as regras são divididas em declarações de variáveis, condições e ações (FILHO, 2000).

Após a escrita da base de regras de produção, os arquivos de regras compostos no JEOPS é pré-compilado em classes Java. Esse arquivo é então compilado juntamente com as demais classes do programa em um compilador padrão Java. As classes compiladas são entregues para execução juntamente com um ambiente de execução JEOPS para então poder ser utilizado. O KEOPS estende a utilização das regras de produção do JEOPS para uso em dispositivos móveis modificando apenas alguns aspectos (ALBUQUERQUE *et al.*, 2002). A primeira modificação ocorre após a compilação padrão do Java, ponto no qual entra em ação um processador de reflexão para suprir a falta de recursos de reflexão na arquitetura JME. A segunda modificação é que ao invés de ser distribuído um ambiente JEOPS junto com o programa final, é distribuído um ambiente de execução KEOPS. A Figura 2.10 apresenta o funcionamento mais detalhado da criação de aplicações KEOPS.

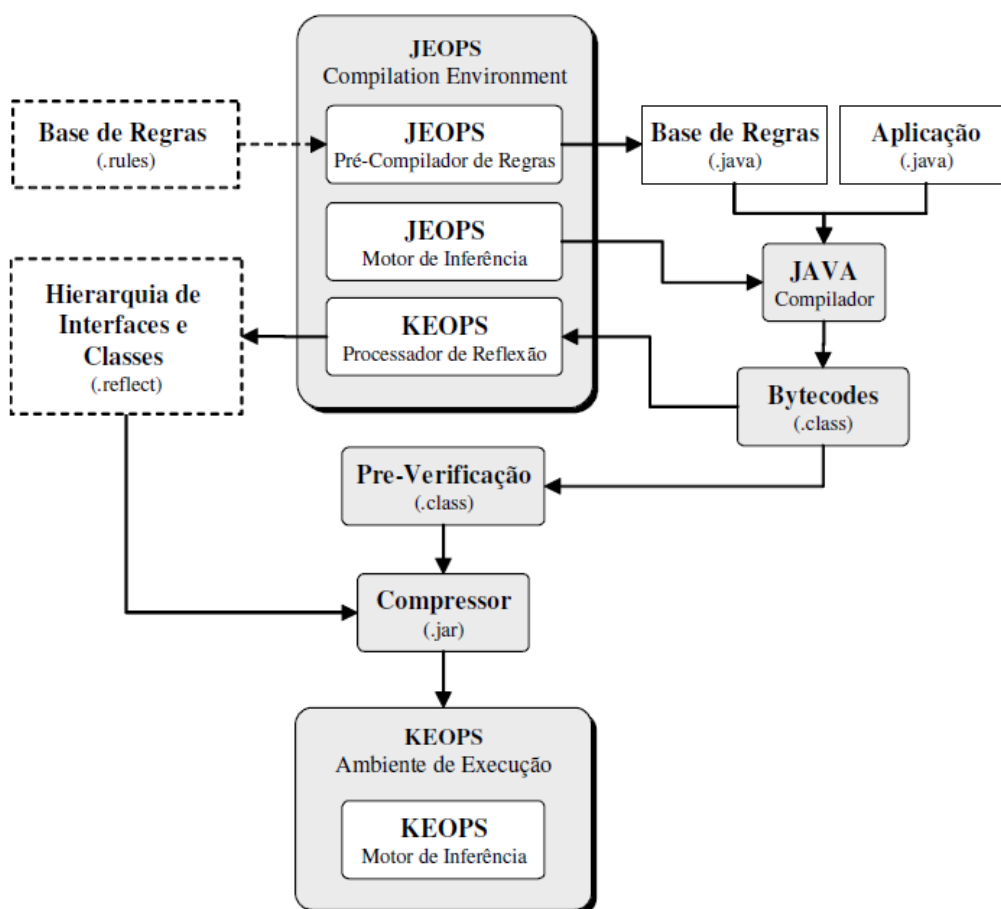


Figura 2.10 - Funcionamento do desenvolvimento de uma aplicação KEOPS (ALBUQUERQUE, 2002, p. 112).

Capítulo

3

Estado da Arte

“Compreender que há outros pontos de vista é o início
da sabedoria.”

Thomas Campbell

A evolução das tecnologias sem fio tem atraído à atenção de todas as áreas do conhecimento. Na medicina, uma ciência que lida diretamente com o bem estar das pessoas e com um montante de informações, não seria diferente. As facilidades obtidas com o uso de dispositivos móveis vêm modificando a aquisição, o armazenamento, a recuperação, o gerenciamento das informações médicas e de saúde (SIQUEIRA, 2007). Ao unir telemedicina com tecnologia móvel é possível viabilizar a interação entre médicos, pacientes e administradoras de saúde, facilitando a globalização da medicina e a democratização do acesso à saúde. Partindo destes princípios, várias pesquisas são realizadas unindo diversos conceitos de tecnologia da informação com aplicações na área médica.

Na área de monitoramento por Redes de Sensores Corporais (RSC), tradução do termo *Body Sensors Network* (BSN), geralmente recorre-se a técnicas de reconhecimento de padrão para extração de informações médicas relevantes dos sensores (JAFARI *et al.*, 2007). Essas técnicas se baseiam em duas tarefas básicas, a primeira cuida da extração de atributos de um objeto e a segunda classifica os objetos em grupos rotulados com base em seus atributos (Figura 3.1). O reconhecimento de padrões utiliza duas abordagens diferentes para classificação da informação: a Estatística e a Estrutural. A primeira abordagem utiliza a teoria da decisão estatística para separar os dados de diferentes grupos com base nas características quantitativas dos dados. A abordagem Estrutural depende de gramáticas de sintaxe para separar os dados de diferentes grupos com base em inter-relações morfológicas presentes nos dados. JAFARI *et al.* (2007), tomando por base a segunda abordagem para reconhecimento de padrões, propõem um modelo de monitoramento usando RSC que se divide em duas fases:

- Na primeira, Unidades de Processamento (UP) captam os sinais fisiológicos do usuário a partir dos sensores sem fio, acoplados no corpo do paciente, fazendo um pré-processamento e uma classificação local dos sinais;
- Na etapa seguinte, um telefone móvel ou um *Pocket PC* coleta os dados das UP e faz uma segunda classificação nos dados, armazenando-os e disparando eventos críticos, quando detectados pelas Unidades de Processamento, para um *gateway* na internet ou para o hospital.

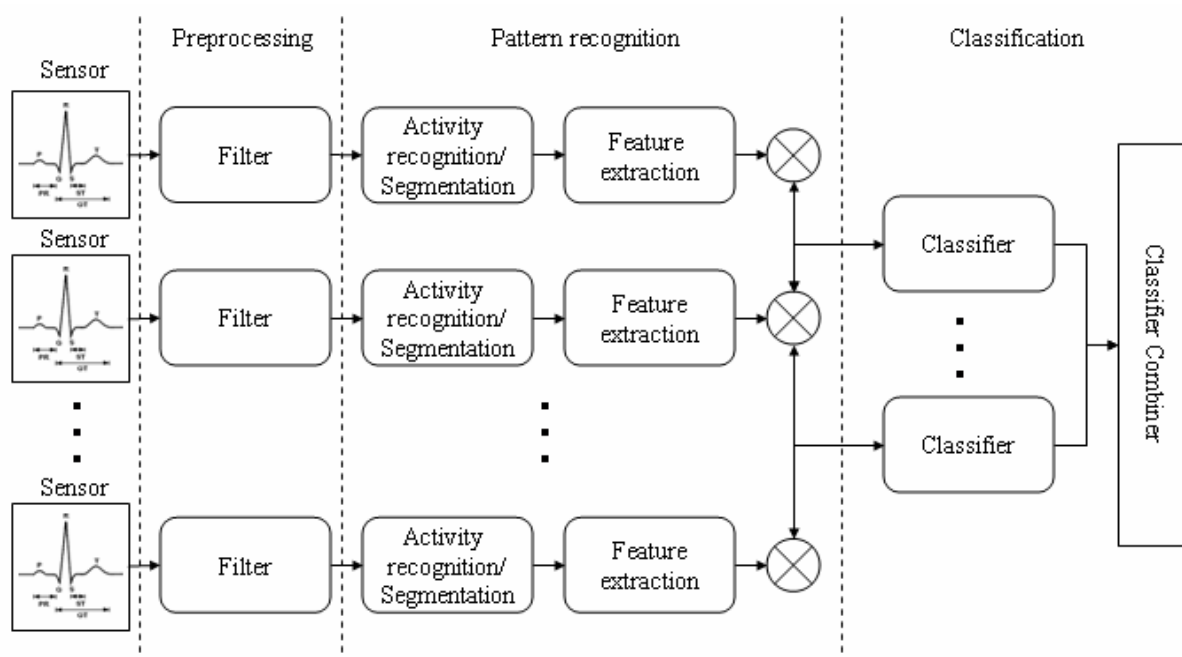


Figura 3.1 - Arquitetura do sistema proposto por JAFARI et al (2007).

A proposta deste modelo de JAFARI *et al.* (2007) é avaliar a movimentação e classificar as atividades físicas dos pacientes monitorados. Com pequenas alterações no modelo, tais como mudança dos tipos de sensores, é possível expandir esta plataforma para observar outras características do comportamento do usuário da RSC e reunir outras informações sobre os hábitos e a saúde do paciente.

ANDREÃO, PEREIRA FILHO e CALVI (2006) também propõem o uso do monitoramento remoto para tentar melhorar a qualidade da hospitalização de pacientes com doenças crônicas cardíacas. Na solução desenvolvida por eles é utilizada uma Unidade Remota (UR) e uma Central de Monitoramento (CM) que podem ser vistas na Figura 3.2. A UR é composta de um dispositivo portátil para aquisição e processamento de sinais eletrocardiográficos e um mecanismo de alarme acionado pelo próprio usuário, alocado próximo ao corpo do paciente adquirindo seus dados fisiológicos, e de um computador remoto, utilizado para receber, armazenar e tratar as informações adquiridas pelo dispositivo

portátil. O computador remoto ainda analisa os sinais do eletrocardiograma (ECG) e gera alarmes para a Central de Monitoramento, através da internet. A CM é composta por um servidor de monitoramento, utilizado para acompanhamento do estado clínico dos pacientes, e uma base de dados conceitual, utilizada para armazenar os dados do paciente e gerar informações para o prontuário eletrônico do paciente.

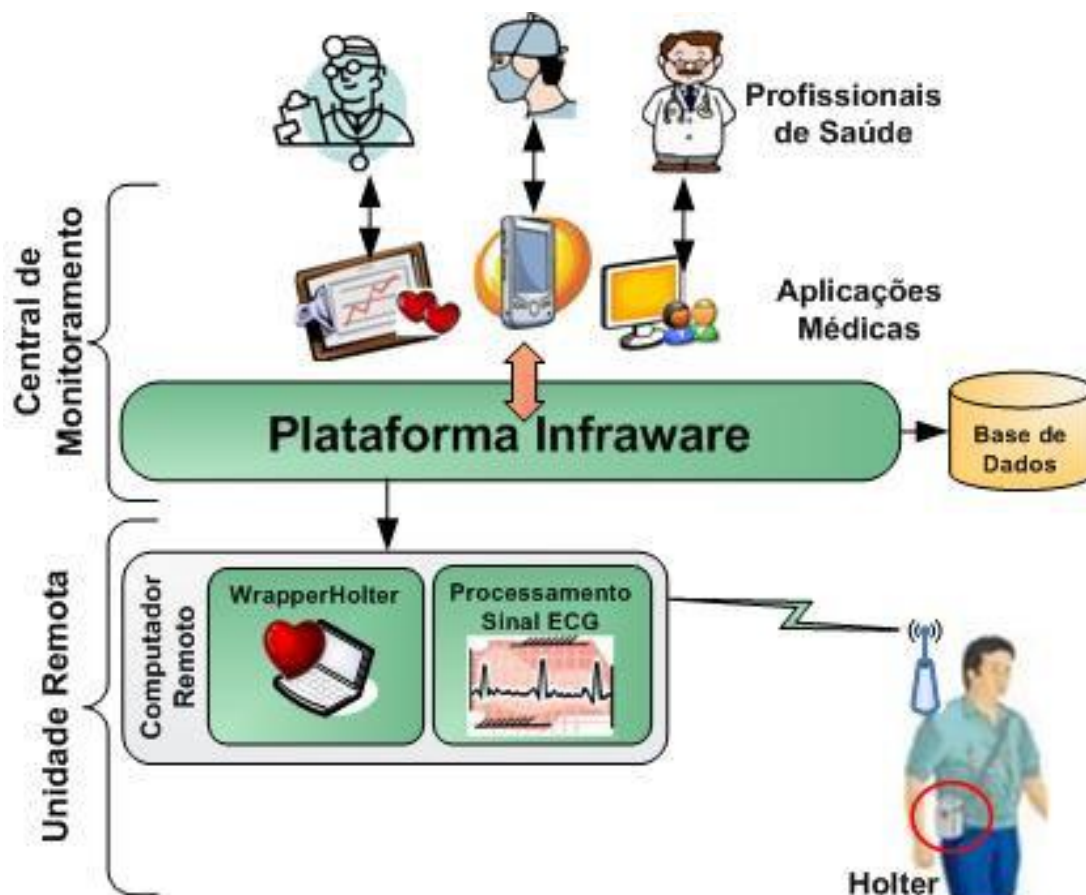


Figura 3.2 - Interação no sistema proposto por ANDREÃO, PEREIRA FILHO e CALVI (2006).

Também interessado no monitoramento de pacientes com doenças crônicas cardíacas ANLIKER *et al.* (2004) desenvolveram um sistema, o AMON (*Advanced care and alert portable telemedical MONitor*), que visa prover uma melhor qualidade de vida aos seus usuários, sem restringir suas atividades e sua mobilidade. O AMON tem como principais características a integração de vários monitores médicos e software de avaliação em um único dispositivo, o posicionamento não padronizado dos sensores de medição, miniaturização do dispositivo a um nível que ele pode ser vestido no pulso (Figura 3.3) e a capacidade de comunicação móvel é embutida no próprio dispositivo, através de um transceptor *on-board* GSM capaz de comunicar o dispositivo com um centro médico. Na Figura 3.4 vemos um esquema de sua arquitetura.



Figura 3.3 - Protótipo AMON (ANLIKER *et al.*, 2004).

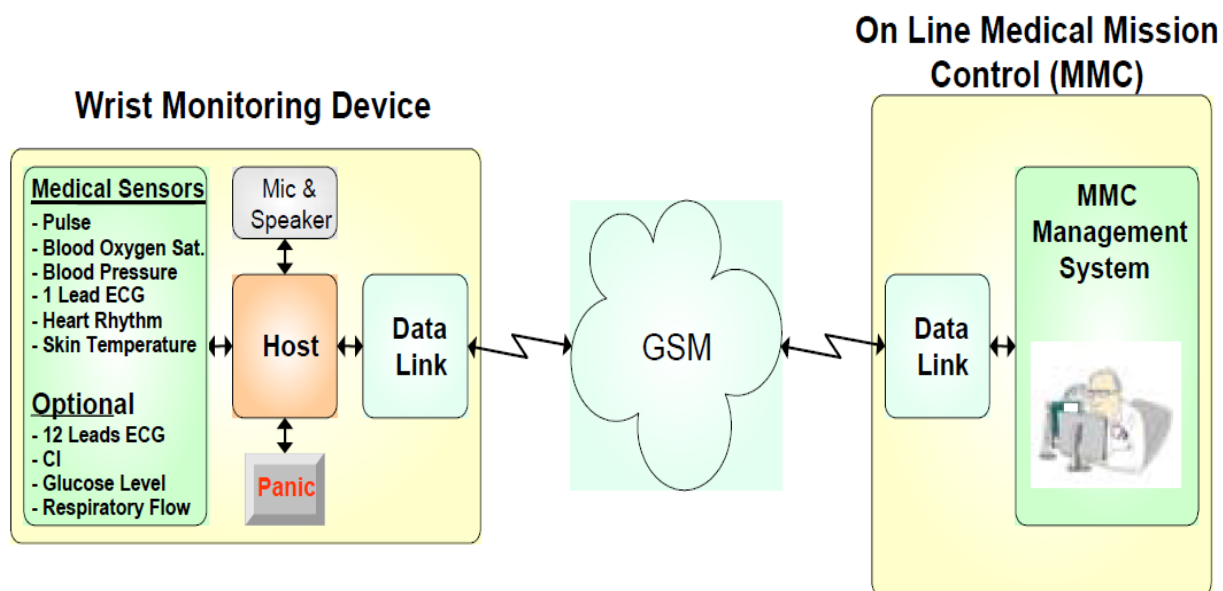


Figura 3.4 - Visão geral do sistema proposto por ANLIKER *et al.* (2004).

MACHADO *et al.* (2008) propõe o desenvolvimento de um sistema que une dispositivos móveis, serviços tipo *web services* e software livre no monitoramento remoto de pacientes (Figura 3.5). O sistema por eles proposto utiliza um monitor cardíaco de tecnologia sem fio para adquirir dados do paciente e enviá-los a um dispositivo móvel com tecnologia

JME. Esse dispositivo recebe os dados, os coloca no padrão XML e os envia por uma rede GPRS a um servidor. No servidor é executado um *web service* que armazena os dados recebidos em um banco de dados, montando a base de dados para o paciente. O servidor também é responsável por analisar o quadro clínico do paciente e, a partir dos dados gerados, enviar mensagens, com as informações sobre o estado de saúde do paciente, para o médico através de SMS ou e-mails.

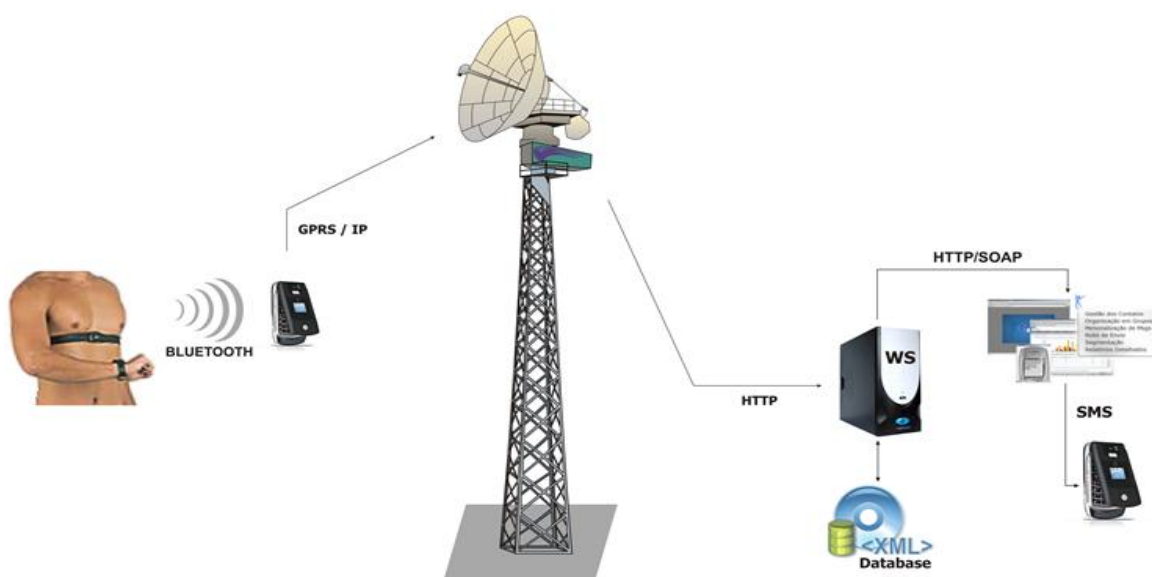


Figura 3.5 - Arquitetura do sistema desenvolvido por MACHADO *et al.* (2008).

LEE *et al.* (2006), também trabalhando com monitoramento remoto de pacientes, desenvolveram uma arquitetura de sistema (Figura 3.6) cujo objetivo é implementar um monitor de pressão sanguínea sob forma de um agente pequeno, leve, portátil e com transmissão Bluetooth. A tecnologia Bluetooth é utilizada para transmissão dos dados adquiridos pela rede para o telefone celular do usuário. O dispositivo móvel, então, armazena os sinais vitais do usuário e os envia para uma central onde os dados são analisados e, dependendo desta análise, o médico é contatado através de seu telefone celular. Esta arquitetura divide o sistema proposto em três partes distintas. A primeira é constituída pelos circuitos analógicos que captam todos os sinais fisiológicos do usuário. A segunda recebe os sinais analógicos, converte-os em digitais e assim exibe-os ao usuário. A terceira, e última parte, é responsável pelas interfaces de transmissão de informações. Nesse modelo, praticamente todas as comunicações com fio foram eliminadas, mas o dispositivo móvel age apenas como um roteador de informações.

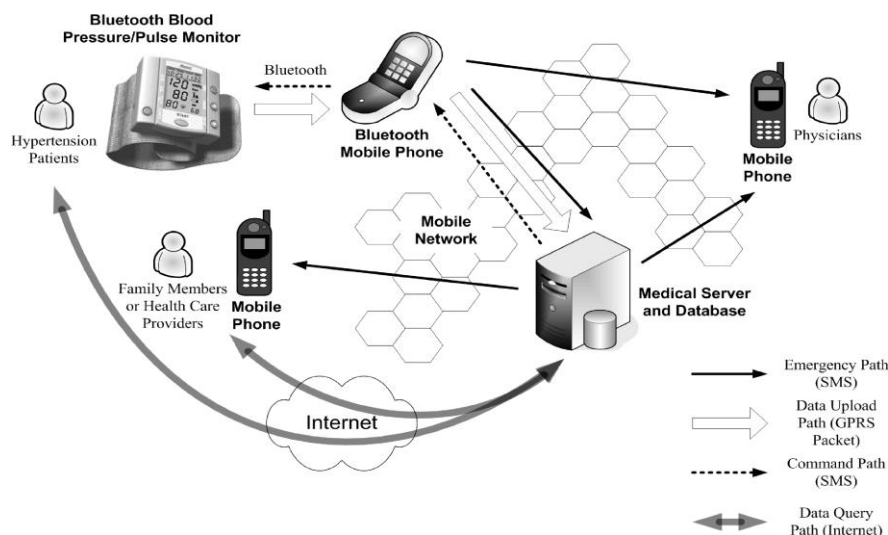


Figura 3.6 - Arquitetura do sistema desenvolvido por LEE *et al.* (2006).

Até agora foram vistas diversas soluções para monitoramento remoto, mas é possível observar que nenhuma utilizou qualquer tipo de inteligência nas técnicas de monitoramento dos pacientes, ou até mesmo em qualquer outra etapa do desenvolvimento de seu sistema. Isto porque a inteligência artificial está começando a ser utilizada na área de telemedicina aos poucos, principalmente devido a questões éticas. Diversos pesquisadores, tal como COSTA (1999), partindo da visão de que agentes inteligentes são entidades autônomas, com diversas características humanas, tais como capacidade de aprendizado, pró-atividade, inferência, capacidade de solucionar problemas heurísticos, agir com base em objetivos e se comunicar, vem apostando no uso de inteligência artificial para auxiliar o trabalho com a grande quantidade de informações médicas e para facilitar a interação e o acesso à saúde. Já é consenso que agentes de *software* podem facilmente manipular um montante de dados do paciente buscando identificar informações relevantes sobre sua saúde em tempo muito mais ágil do que um ser humano.

Pesquisadores, como LIEBERMAN e MASON (2002), acreditam que agentes colaborativos, ou um time de agentes de *software*, podem ser utilizados para analisar dados e resolver problemas a partir de diversos pontos de vista e com capacidades de múltiplas competências. Com isso, os grupos de agentes inteligentes aumentam a velocidade de integração e priorização de diversos dados em larga escala através de complexos sistemas de distribuição de informação. Essa colaboração entre agentes, que é ressaltada no diagrama de fluxo de informações na Figura 3.7, auxilia médicos, pacientes e administradoras de saúde a lidar com o problema da sobrecarga de informações fruto da explosão de informações médicas e da complexidade dos sistemas de computadores.

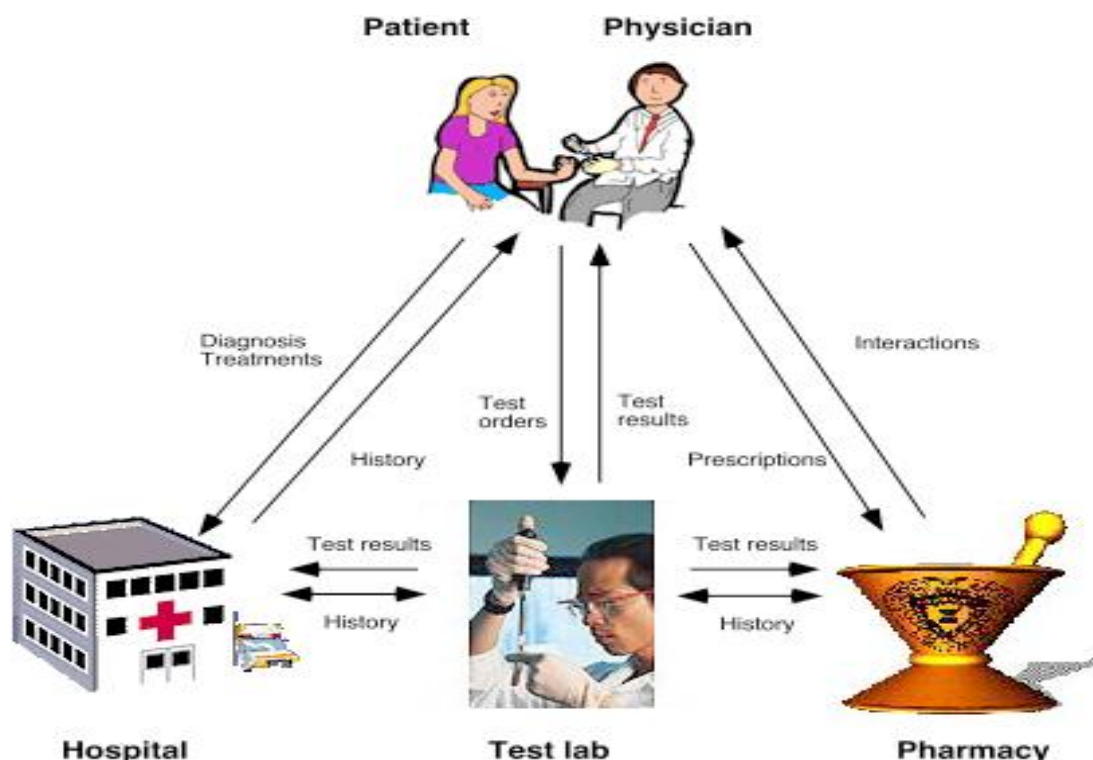


Figura 3.7 – Diagrama de fluxo de informações entre os diferentes atores de um sistema médico, organizado por LIEBERMAN e MASON (2002).

Outros sistemas, como o proposto por MABRY *et al.* (2003), propõem o uso dos agentes inteligentes para o monitoramento e até o diagnóstico de pacientes, agindo como Sistemas de Apoio à Decisão (SAD) médica. Segundo esses autores, os agentes passam a ter o papel de realizar diagnósticos com base em um conjunto complexo e dinâmico de informações, fazer análises contínuas e dar suporte às decisões de intervenção, provendo aos especialistas médicos recursos auxiliares preciosos.

Já HERBERT *et al.* (2006) discutem a construção de uma arquitetura de integração de agentes móveis para aplicações médicas que usam redes de sensores sem fio. Para isto, duas plataformas de agentes inteligentes, a Jade e a Agilla, são integradas (Figura 3.8). A primeira plataforma é utilizada para dispositivos que tenham mais recursos de processamento. Por exemplo, servidores, PCs, PDAs e telefones celulares de melhor desempenho. A Agilla é utilizada para plataformas com recursos mais restritos, como os sensores. A arquitetura proposta pelos autores, chamada *Data Management System (DMS)*, desenvolve uma maneira de prover o gerenciamento de dados em uma rede de sensores sem fio do paciente que seja eficaz e efetiva, buscando garantir uma boa assistência ao usuário.

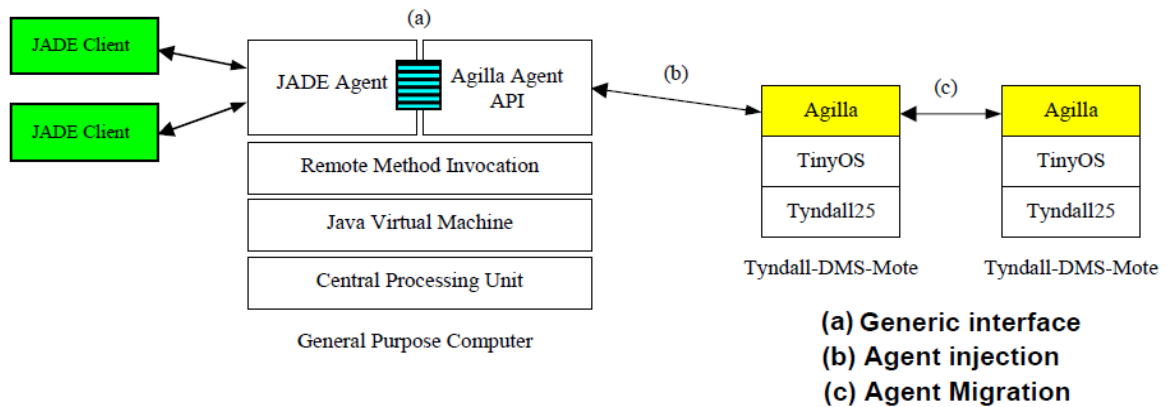


Figura 3.8 - Integração entre JADE e Agilla proposto por HERBERT *et al.* (2006).

Na área de monitoramento de pacientes remotamente, alguns pesquisadores vem desenvolvendo sistemas *home care* utilizando inteligência artificial para produzir regras em conjunto com especialistas médicos para identificação de situações anormais no paciente. Um exemplo é o sistema apresentado por COPETTI (2010), no qual sensores de ambiente, distribuídos pela casa do paciente, e sensores de sinais fisiológicos, utilizados junto ao corpo do paciente, captam dados e um computador recebe e processa esses dados determinando as condições de saúde do paciente (Figura 3.9). Para o diagnóstico da situação do paciente no computador, um programa utiliza variáveis *fuzzy* e um conjunto de regras de produção. O computador, além de fazer uma análise da situação do paciente, funciona como um gateway entre a residência do paciente, os provedores de saúde e os familiares do paciente.

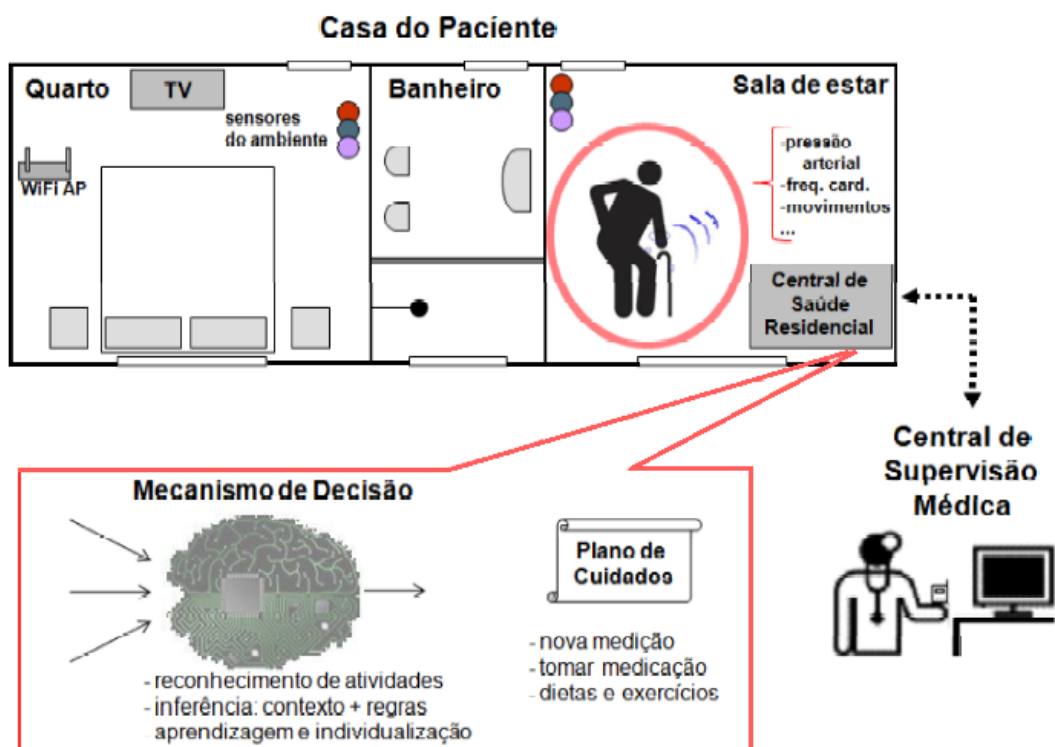


Figura 3.9 - Arquitetura do sistema desenvolvido por COPETTI (2010).

Diversos outros autores abordam o uso de tecnologia para monitoramento de pacientes (CARVALHO, 2005; JOVANOVA *et al.*, 2001; MEA, 2001; MENEZES JÚNIOR, J. V. *et al.*, 2011; PATTICHIS, C. S. *et al.*, 2002; VOSKARIDES, S. *et al.*, 2002; PIZARRO, P. J. C. *et al.*, 2001; XIAO; CHEN, 2008). E várias pesquisas unem conceitos de IA, telemedicina e dispositivos móveis, mas geralmente elas concentram-se nos aspectos de suporte da rede ou com mecanismos de segurança e privacidade, ver Tabela 3.1.

Independentemente da natureza da pesquisa, pode-se dizer que a maioria dos trabalhos utiliza sempre um dispositivo sem inteligência, isto é, apenas um dispositivo que só replica a informação recebida, intercomunicando a rede de sensores sem fio e um computador. Quando se discute o cuidado com pacientes de risco no cenário *home care* esse dispositivo sem processamento algum dos dados pode inexistir, pois a rede de sensores quase sempre se comunica com o computador na residência do paciente, que é um centro de processamento e análise de informações remotas. Mas quando a mobilidade é o principal componente para o sucesso do monitoramento do paciente, o dispositivo móvel, na maioria das vezes um telefone celular, ocupa o lugar do nó sem inteligência, agindo como um simples repositório e roteador de informações. No entanto, o dispositivo móvel pode ser programado para ser um elemento ativo, para ter um comportamento autônomo e ser capaz de tomar decisões de acordo com sua base de conhecimentos.

Tabela 3.1 - Síntese das características dos sistemas encontrados no estado da arte.

	<i>Body Sensors Network (JAFARI et al., 2007)</i>	<i>TeleCardio (ANDREÃO et al., 2006)</i>	<i>AMON (Anliker et al., 2004)</i>	<i>Machado et al. (2008)</i>	<i>Sistema proposto por Lee et al. (2006)</i>	<i>DMS (HERBERT et al., 2006)</i>	<i>H-SAUDE (COPETTI, 2010)</i>	<i>EMERGE (XIAO; CHEN, 2008)</i>
Objetivo	Classificação de movimentos físicos	Monitoramento remoto de pacientes cardíacos	Monitoramento remoto de pacientes cardio-respiratórios	Monitoramento remoto de pacientes	Monitoramento remoto de pacientes hipertensos	Gerenciamento de dados flexível e efetivo em redes de sensores sem fio de pacientes	Monitoramento remoto ubíquo e inteligente	Monitoramento de idosos
Variáveis consideradas	Temperatura, pressão arterial e condutância do nível da pele	Eletrocardiograma (ECG)	Taxa de pulso, oxigenação, pressão arterial, ECG, temperatura e acelerômetros	Ritmo cardíaco, temperatura corporal e pressão sanguínea	Pressão sanguínea, taxa de pulso, ECG	ECG, pressão sanguínea e taxa de pulso	Pressão e frequência cardíaca, atividade do paciente e temperatura ambiente	Movimento, posição horizontal, temperatura corporal e fotopletismografia
Tipo de sistema (Móvel ou Home care)	Móvel (Unidades de Processamento (UP) Local e Pocket PC, ou Celular)	Híbrido	Móvel (Dispositivo próprio)	Móvel (Celular)	Móvel (Celular)	Híbrido (Computadores e PDAs)	<i>Home care</i>	<i>Home care</i>
Mecanismo de Inteligência remoto	Reconhecimento de padrões nas UP	Inexistente	Classificação com base em pré-definição	Inexistente	Classificação com base em pré-definição	Uso de ontologias e múltiplos agentes	Ontologias e lógica Fuzzy	Reconhecimento de padrões
Comunicação com Central Médica	Internet	Através de um computador	SMS/GPRS e Internet	GPRS (GSM e TDMA)	SMS/GPRS	Não explicitado	Não explicitado	Não explicitado
Linguagem de programação	Não explicitada	Não explicitada	Java	Java	Java	Java	Não explicitada	Não explicitada

Capítulo

4

Procedimentos Metodológicos

Toda a ação é designada em termos do fim que se procura atingir.

Niccolò Machiavelli

Neste capítulo são descritos os passos necessários à concepção, projeto e elaboração do sistema de monitoramento remoto preventivo e inteligente proposto neste trabalho. As seguintes atividades foram realizadas para alcance dos objetivos propostos no primeiro capítulo:

- Levantamento bibliográfico: busca e leitura de artigos e livros relacionados à área de telemedicina, em especial os que traziam como tema central o monitoramento remoto preventivo de saúde, com a filtragem dos mais relevantes⁹;
- Projeto de um sistema de monitoramento de saúde inovador: definição do escopo e concepção do sistema, levando em consideração os aspectos do estado da arte e os possíveis avanços em relação a ele;
- Implementação do sistema: modelagem, design e construção dos componentes integrantes do sistema *MonitorPrevIntel* que atendam aos objetivos propostos neste trabalho;
- Validação: execução, testes e validação dos componentes que fazem parte do sistema implementado para garantir sua integração e seu bom funcionamento.

As atividades de desenvolvimento do sistema tomaram por base o modelo espiral, que é um processo de software cíclico, de abordagem evolucionária e que tem, como principal

⁹ Foram pesquisados artigos nas principais bases científicas da área de Computação, entre elas IEEE Xplore (<http://ieeexplore.ieee.org/>), ACM Digital Library (<http://dl.acm.org/>), SciELO – Scientific Eletronic Library Online (<http://www.scielo.org/>), Portal de Periódicos da Capes (<http://www.periodicos.capes.gov.br/>). Os artigos foram selecionados como relevantes por conter características mais próximas aos objetivos deste trabalho, das quais se destacam: utilização de mecanismo de inteligência, linguagem de programação orientada a objeto, utilização de sensores para dados fisiológicos, utilização de tecnologia móvel e comunicação remota.

destaque, a orientação a riscos (BOEHM, 1988). Cada ciclo deste modelo de desenvolvimento é dividido em quatro fases, também chamadas de tarefas. Na primeira dessas são determinados os objetivos, soluções alternativas e restrições. Na segunda fase as alternativas da fase anterior são avaliadas e os riscos são identificados e solucionados. A terceira fase está relacionada ao desenvolvimento em si, o qual inclui o design, especificação, codificação e verificação. Na quarta e última fase, é realizada uma revisão das tarefas anteriores e o planejamento do próximo ciclo (SOMMERVILLE, 2007). Ao usar este modelo pode-se optar pelo uso de sua versão completa (Figura 4.1) ou escolher quantos ciclos forem necessários, bem como por qual modelo de desenvolvimento utilizar na terceira fase de desenvolvimento.

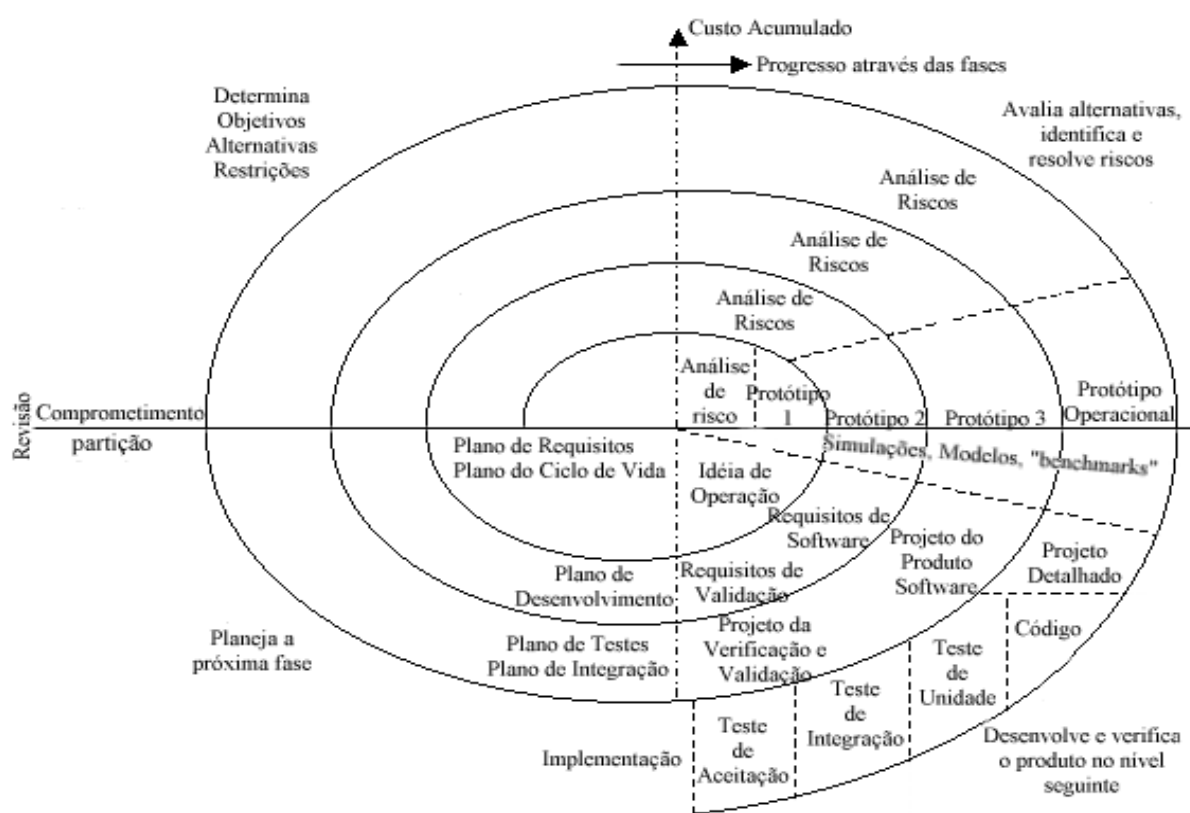


Figura 4.1 - Modelo Espiral Completo (SOMMERVILLE, 2007).

A escolha deste modelo de desenvolvimento de software para este trabalho fundamentou-se na necessidade de um modelo que atendesse suas características evolutivas, bem como pela sua abordagem ao lidar com os riscos. Vale ressaltar que os riscos são compreendidos como situações adversas que surgem ao longo do desenvolvimento de um sistema (LEITE, 2007), como: falta ou mudança de tecnologia, ferramentas que não podem ser utilizadas, problemas com a equipe de desenvolvimento, entre outras.

No desenvolvimento do *MonitorPrevIntel*, cada ciclo envolveu a implementação de um componente funcional do sistema. Ao final no último ciclo, foi construído o agente

inteligente que integrou cada um dos componentes desenvolvidos anteriormente. Apesar de uma especificação geral do *software* ter sido elaborada desde o primeiro ciclo, a cada ciclo a especificação necessitou de alterações devido aos riscos. Os principais riscos detectados no desenvolvimento deste trabalho foram: tecnologia de geração de regras de produção que não atendia aos requisitos do sistema de processamento completamente embutido; limitações das tecnologias utilizadas para desenvolvimento do *software*. Devido a estes riscos foi interessante a utilização da abordagem evolutiva para adaptação do desenvolvimento aos objetivos propostos.

Além de um processo de desenvolvimento, a elaboração de um sistema também envolve estudo e uso de diferentes tecnologias que permitam sua construção. Seja elaboração de sua especificação, seja implementação funcional do sistema. Nas subseções a seguir são apresentadas as tecnologias envolvidas na realização deste trabalho.

4.1 Modelagem

Algumas tecnologias permitem a interação entre o processo de desenvolvimento e a implementação do sistema, facilitando sua compreensão. Assim é a UML, utilizada neste trabalho, que serve para modelagem gráfica do sistema e de seus componentes, melhorando sua legibilidade. Estes modelos gráficos são chamados de diagramas e auxiliam na representação da parte estrutural, comportamental e arquitetural do sistema (OMG, 2011).

Para modelagem dos diagramas UML, é necessário o uso de alguma ferramenta que permita a edição dos mesmos. Optou-se então pelo uso do *Astah Community*, pois esta é uma ferramenta gratuita que traz aos desenvolvedores grandes recursos e boa praticidade na elaboração dos diagramas (CHANGE, 2012).

4.2 Implementação

A linguagem de programação escolhida para implementação do *MonitorPrevIntel* foi a linguagem Java que se caracteriza por ser de escrita simples, orientada a objetos, distribuída, robusta, segura, com alto desempenho, *multi-thread* e como propunha seu slogan, “Write

once, run anywhere”¹⁰, interpretada, dinâmica e independente de arquitetura (CARDOSO, 2007 apud GOSLING, 1995). Essa linguagem, padronizada sob as diretrizes e as revisões da *Java Community Process*¹¹ (JCP), é dividida em três ou quatro¹² plataformas, as quais são conjuntos de funções, bibliotecas e recursos, que diferem de acordo com as características do programa a ser desenvolvido. Como o aplicativo desenvolvido nesse trabalho requer o uso de uma linguagem de programação com recursos que possibilitem a mobilidade, a portabilidade, a persistência de dados, o estabelecimento de conexão SMS e suas demais características, optou-se por utilizar a plataforma *Java Mobile Edition* (JME).

A JME é voltada para o desenvolvimento de aplicações para dispositivos com poucos recursos de potência e processamento (JCP, 2012). Em outras palavras, essa plataforma tem como objetivo o desenvolvimento de aplicativos para dispositivos com restrições de memória, potência e capacidade gráfica, conectividade intermitente, conceitos de sistema de arquivos divergentes e diferentes sistemas operacionais (HAYUN, 2009). Entre os quais estão: os *set-top boxes*, as TVs digitais, PDAs, impressoras, componentes de automação residencial e os telefones móveis.

A JME tem um conjunto de recursos e bibliotecas, da linguagem Java, reduzido para poder atender os requisitos de menor capacidade computacional dos dispositivos ao qual se destinam, provendo uma uniformidade de recursos independentemente do fabricante desses. Assim, precisou-se remover muitos recursos para adequar a linguagem aos dispositivos de baixa capacidade computacional, alguns desses recursos são:

- Finalização automática de objetos (cláusula *finally*);
- *Java Native Interface* (JNI);
- Reflexão;
- Serialização de objetos;
- Grupos de *threads* e controle avançado das mesmas;
- Carregadores de classes definidos pelo usuário.

O JME é dividido em dois grupos com focos específicos, um voltado para dispositivos de maior capacidade funcional que geralmente são fixos, os *Connected Device Configuration*

¹⁰ Esta expressão significa “Escreva uma vez, execute em qualquer lugar”. Faz alusão à compilação dos programas Java para um código intermediário que é executado por uma máquina virtual, assim uma vez compilados os programas podem ser utilizados em qualquer plataforma, tornando-se portáteis.

¹¹ Programa que tem como objetivo desenvolver e revisar as especificações, implementações de referência e conjuntos de testes da tecnologia Java (JCP, 2012), criado em 1998 a partir de um consórcio entre as maiores empresas de aparelhos eletrônicos (ALBUQUERQUE, 2002).

¹² Três plataformas da tecnologia Java: *Java Enterprise Edition* (JEE), a *Java Standard Edition* (JSE) e *Java Micro Edition* (JME). Ainda há quem considere a especificação *Java Card*.

(CDC), e outro voltado para dispositivos de menor capacidade funcional que geralmente são móveis, os *Connected Limited Device Configuration* (CLDC). A arquitetura da plataforma JME, vista na Figura 4.2, é dividida quatro partes: máquina virtual, configuração, perfil e pacotes opcionais.

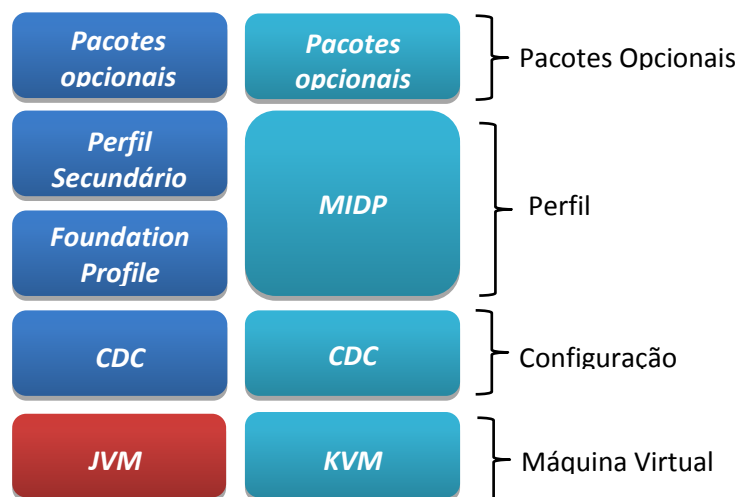


Figura 4.2 - Arquitetura da plataforma JME.

Na Figura 4.2 é possível observar os dois grupos da arquitetura JME, um utiliza uma JVM e atende aos requisitos da CDC e o outro utiliza a KVM e atende aos requisitos da CLDC. Para este trabalho o grupo de interesse é este último, por ser voltado ao desenvolvimento para dispositivos de menor capacidade computacional. Gatto (2007) reúne uma lista de requisitos mínimos necessários aos dispositivos que utilizam o perfil MIDP, apresentados na Tabela 4.1

Tabela 4.1 - Requisitos mínimos para os dispositivos MIDP.

Requisitos Mínimos	
Hardware	Display: 96x54, 1 bit de profundidade e 1:1 formato do pixel
	Input: teclado de uma mão, de duas mãos ou QWERTY
	Memória ROM: 128Kb para MIDP e 8Kb para dados de aplicação
	Memória RAM: 32Kb para executar o Java
	Rede: Duplex, sem fio, intermitente e largura de banda limitada
Software	Som: mono 8KHz
	Recursos para executar a KVM
	Tratamento de exceções
	Processamento de interrupções
	Escrita de elementos gráficos <i>bitmap</i> na tela
	Acesso de leitura e escrita à rede sem fio
	Mecanismo para capturar entrada de um <i>input device</i>
	Recursos para ler e gravar em memória não volátil para suporte de dados persistentes

Os pacotes e APIs¹³ opcionais servem para suprir a necessidade de outras funcionalidades que não fazem parte do conjunto padrão de recursos do JME, disponibilizando recursos muito específicos, tais como: conectividade à base de dados, mensagens multimídia, gráficos 3D, *Web Services*, localização, capacidade de realizar chamadas telefônicas, acesso à agenda de contatos e à lista de arquivos, citados em Lizakoski (2008), Regueira (2006) e Hayun (2009). Os pacotes opcionais e APIs podem ser proprietárias aos fabricantes dos dispositivos¹⁴, podem ser abertas e desenvolvidas por programadores independentes ou podem ser desenvolvidas sob a supervisão do JCP e disponibilizadas para uso de qualquer fabricante.

Abaixo são listadas as principais APIs utilizadas durante a implementação deste trabalho. Algumas são opcionais, outras já fazem parte do perfil ou da configuração:

- *Bouncy Castle Crypto*: conjunto de classes de código aberto que implementa vários algoritmos de criptografia, porém mais que isso, é uma implementação *cleanroom*¹⁵ da *Java Cryptography Extension* (JCE) 1.2.1 que possui uma distribuição específica para o JME (BOUNCY, [20??]);
- *Record Management System* (RMS): conjunto de classes criado para permitir a persistência de dados em dispositivos móveis através de *Records Store*, que já faz parte do pacote padrão da plataforma JME a partir da versão MIDP 1.0;
- *Wireless Messaging API* (WMA): conjunto de classes que permite o estabelecimento de conexões para envio de mensagens através do padrão GSM, foi desenvolvido desde a implementação do CLDC 1.0.

Uma aplicação desenvolvida para dispositivos móveis é chamado de MIDlet que pode rodar em qualquer aparelho que tenha a KVM, a CLDC e o MIDP. O ciclo de vida de um MIDlet é composto de três estados, apresentados na Figura 4.3, que são gerenciados no dispositivo móvel pelo *Application Management Software* (AMS) (MURCHOW, 2004). O AMS administra o ciclo de vida de um MIDlet e fornece o ambiente em tempo de execução, impondo segurança, permissões, fornecendo classes do sistema do dispositivo e agendamentos (MAREJKA, 2005).

¹³ Acrônimo para *Application Programming Interface* que são um conjunto de bibliotecas e recursos utilizados para determinada função.

¹⁴ Os aplicativos que utilizam estas funcionalidades perdem sua portabilidade para outros dispositivos, ficando limitados aos aparelhos com este atributo.

¹⁵ É um processo de desenvolvimento de software caracterizado por alta confiabilidade sob controle estatístico de qualidade e tem como objetivo desenvolver sistemas que apresentem zero falhas na utilização (FOREMAN, 2005).

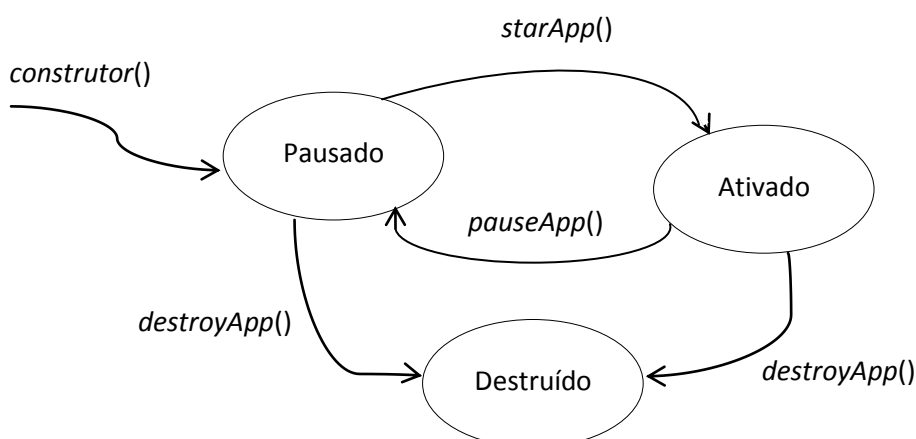


Figura 4.3 - Ciclo de vida de um MIDlet.

Existem diversas ferramentas que permitem criar um ambiente adequado para o desenvolvimento e os testes de aplicativos para dispositivos móveis. De maneira geral este ambiente é alcançado através do uso de um kit de desenvolvimento Java (*Java Development Kit*, JDK), um editor de texto para codificação e um emulador para o aplicativo móvel. Neste trabalho optou-se pelo uso da IDE NetBeans 6.9.1 como editor de texto, como JDK foi utilizada a versão 1.6.0 *update* 21 e o emulador Wireless Toolkit 2.5.2 (WTK). O programa foi desenvolvido para dispositivos que usem configuração CLDC 1.1 e perfil MIDP 2.0.

Para fazer a criptografia das mensagens de texto a serem enviadas pelo MIDlet optou-se por utilizar a cifra de bloco AES-128. Esta criptografia divide a informação a ser criptografada em blocos de 128 bits e utiliza um vetor de inicialização do algoritmo para ir permutando informações e criptografando a informação com base na chave que neste programa terá tamanho de 128 bits.

Definidas as ferramentas e o processo de desenvolvimento de software a serem utilizados na construção do *MonitorPrevIntel*, deu-se início a sua implementação que será abordada no capítulo 5.

Capítulo

5

MonitorPrevIntel

“Se queremos progredir, não devemos repetir a história,
mas fazer uma história nova.”

Mahatma Gandhi

Após a explanação dos procedimentos metodológicos e a descrição das tecnologias utilizadas para elaboração do aplicativo apresentado neste trabalho, este capítulo apresenta o aplicativo inteligente para telefones móveis para monitoramento remoto preventivo de pacientes com doenças cardiovasculares, o *MonitorPrevIntel*. Esse sistema foi construído com o intuito de validar a aplicabilidade dos objetivos propostos na seção 1.2 do primeiro capítulo.

De maneira sucinta, o aplicativo embutido no celular lerá os dados relativos aos sinais vitais do paciente, usuário do sistema, confrontá-los com a base personalizada de regras de produção e então, a partir da(s) ação(ões) ativada(s) através das regras, notificar o usuário e a central de supervisão de saúde sobre situações de alertas e emergências médicas.

Ao longo deste capítulo será apresentado, com maiores detalhes, o funcionamento do *Monitor Intelligent Prevention*. Deste modo, serão descritas suas características gerais, sua arquitetura e seus módulos, o fluxo de execução, os diagramas e demais elementos que fizeram parte do desenvolvimento desse aplicativo.

5.1 *Características Gerais*

O objetivo principal deste sistema é realizar o monitoramento de saúde inteligente que mantenha o paciente supervisionado, durante a realização de suas atividades diárias, de maneira ubíqua. A ubiquidade é proporcionada através do uso do telefone celular como *hardware* para este sistema. Este dispositivo também facilita o acesso ao público alvo e reduz

os custos com a compra de equipamento móvel para a monitoração. Além do que, a possibilidade da utilização de inteligência embutida¹⁶ ao aparelho celular permite sua utilização como um elemento ativo no acompanhamento da saúde.

Para realização deste objetivo o *MonitorPrevIntel* é um aplicativo desenvolvido para executar em aparelhos de telefonia móvel, com capacidade de execução de aplicativos Java, de maneira independente da ativação do usuário, capaz de inicializar chamadas telefônicas, enviar mensagens SMS, armazenar os sinais vitais do paciente e demais mensagens que se façam necessárias ao acompanhamento da saúde de seu usuário. Além disso, com esse aplicativo foi desenvolvido um mecanismo de inferência de regras de produção que as manipula e as executa internamente ao aparelho móvel, avançando no estado da arte à medida que isso permite a modificação da base de regras no próprio aparelho móvel, podendo ser feito, inclusive, em tempo de execução.

É importante ressaltar que, por estar lidando com a saúde de pacientes, a inteligência no aplicativo é utilizada de maneira restrita, ficando limitada a interpretação da base de regras de produção específicas para o paciente e detecção de situações anormais de saúde. Assim, apesar do avanço em relação à possibilidade de modificação da base de conhecimento, esta ação não é executada pelo aplicativo ficando a cargo de um profissional da área de saúde, especificamente um médico, construir as regras de produção.

5.2 Arquitetura

Uma visão em alto nível do *MonitorPrevIntel* divide-o em dois módulos, apresentados na Figura 5.1. O primeiro, chamado Módulo de Inferência (MI), é responsável por interpretar os dados dos sinais vitais de acordo com as regras de produção contidas na base de conhecimentos (*Knowledge Base*, KB) e gerar mensagens informando eventuais situações de risco à saúde do usuário. O segundo, chamado Módulo Relator (MR), é responsável por enviar as mensagens geradas no MI a partir de SMS.

¹⁶ O termo embutido refere-se ao processamento de inteligência artificial exclusivamente no aparelho celular.

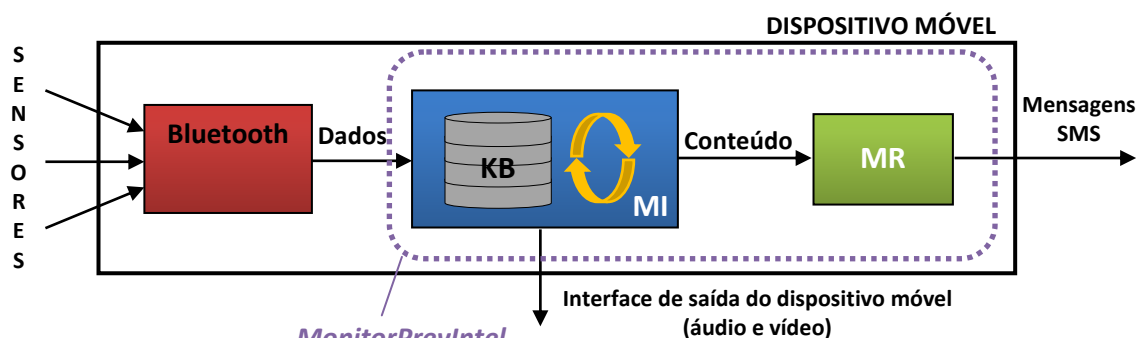


Figura 5.1 - Visão geral da arquitetura do sistema *MonitorPrevIntel*.

O MI lê os dados de sinais vitais, armazena-os em uma base de dados e gera informações a partir da análise dos mesmos. Essa análise é realizada pelo componente inteligente do aplicativo, o sistema de produção, composto por uma base de dados, chamada de memória de trabalho, que contém o estado atual dos sinais vitais do usuário, uma base de conhecimentos, com as regras de produção personalizadas, e um interpretador de regras, chamado de motor de inferência. O MR é acionado através do MI e é utilizado estritamente para reportar para uma central médica as informações geradas pelo *MonitorPrevIntel*, através de mensagens SMS. Este módulo quebra a mensagem em pacotes, criptografa-os e então os envia.

Entretanto, os dois módulos, apesar de dividirem o sistema por funcionalidade, não dizem muita coisa a respeito do desenvolvimento do *Monitor Intelligent Prevention*. Para compreender como o sistema foi construído e como ele funciona, sua arquitetura será apresentada com base nos componentes estruturais e funcionais da mesma. Nas próximas seções serão descritos o sistema de produção, as classes do sistema e sua base de dados.

5.2.1 Sistema de Produção

O sistema de regras de produção é a parte do *software* responsável pela análise do estado de saúde do paciente, utilizando a representação do conhecimento do médico especialista personalizada para aquele determinado usuário. É esse módulo que contém a KB formada pelas regras de produção que representam os conhecimentos do especialista (base de conhecimentos) e a inteligência necessária a manipulação desse conhecimento (motor de inferência) aliado aos dados fisiológicos atuais do usuário (memória de trabalho).

Esse sistema é o componente central do Módulo de Inferência, sendo implementado através da classe principal do sistema, a classe *AgentMidlet*. Inicialmente pensou-se em utilizar a ferramenta KEOPS para geração do mecanismo de inferência, mas como seria necessária a compilação prévia, em um computador, da base de regras para cada nova base gerada, esse processo inviabilizaria a atualização da base de regras do usuário e o processamento unicamente no dispositivo móvel. Em vista disso, foi decidido criar um mecanismo de inferência próprio para este trabalho. Nas seções a seguir serão descritos, de maneira mais detalhada, cada um dos componentes desse sistema de produção.

5.2.1.1 Memória de Trabalho

A memória de trabalho é composta pelos dados que servirão de alimentação ao sistema de produção. Nela estão os dados que serão interpretados de acordo com as regras da base de conhecimentos. Para uso neste trabalho foram escolhidos os dados de: frequência cardíaca (*heartRate*), pressão sanguínea sistólica (*systolicBloodPressure*) e diastólica (*diastolicBloodPressure*), frequência respiratória (*respiratoryRate*), oxigenação do sangue (*bloodOximetry*) e temperatura (*temperature*). Isto porque são os dados que apresentam maior importância para o acompanhamento de saúde de pessoas com doenças cardiovasculares, objeto de estudo do *MonitorPrevIntel*. Como esses dados indicam as características de saúde do usuário do sistema em determinado momento, eles também são gravados na base de dados do sistema, para que sejam utilizados futuramente na análise do histórico de saúde do paciente.

5.2.1.2 Motor de Inferência

O motor de inferência é um mecanismo de engenharia interpretativa das regras de produção. Suas funções são: disparar as regras, selecionar regras que satisfazem a situação atual, resolver conflitos e ativar ações. Disparar as regras significa começar a leitura da base de conhecimentos e comparar suas precondições com a memória de trabalho, interpretando-as. As regras que forem acionadas são selecionadas para execução. Quando mais de uma regra

é selecionada surgem então alguns conflitos: Qual regra deve ser executada? Qual deve ser a ordem de execução? Como deve ser estabelecida essa ordem?

Para resolver essas questões o motor de inferência implementa um mecanismo de resolução de conflitos, que deve decidir quais das regras selecionadas deverão ter suas ações executadas e em que ordem isso deverá ocorrer. Como este trabalho utiliza regras de conhecimento médico que detectam possíveis situações de risco à saúde do usuário quando diante de uma situação de conflito o sistema executará a ação de todas as regras ativadas. A ordem prioritária de cada regra, que deverá ser elaborada pelo especialista da área médica, é definida pela sua posição na base de conhecimentos, a primeira regra a ser lida é a que tem maior prioridade sobre as demais e assim sucessivamente até chegar à última regra a ser lida que deve ser a de menor precedência.

Cada ação é executada logo após a interpretação das precondições de sua regra permitindo que a comunicação de uma anomalia nos sinais fisiológicos a uma central médica ocorra de modo mais rápido. Em outras palavras, assim que uma regra é acionada o motor de inferência a seleciona e ativa sua ação e, então, volta a interpretar as demais regras. A ativação da ação corresponde à sua execução.

As ações disparadas pelo *MonitorPrevIntel* são: *normal* que detecta que as condições de saúde do usuário estão dentro da normalidade, *alert* que detecta alguma anomalia nos sinais vitais do usuário e envia um aviso de alerta a central de supervisão médica e *emergency* que detecta alguma anomalia mais grave nos sinais vitais do usuário e envia um aviso de emergência a central de supervisão médica, juntamente com pedido de assistência. Cada uma dessas ações apresenta comportamento próprio e sua descrição mais detalhada será apresentada na seção 5.2.2 Classes do *MonitorPrevIntel*.

5.2.1.3 Base de Conhecimentos

No aplicativo *MonitorPrevIntel* a base de conhecimentos é armazenada em um arquivo texto nomeado *RuleBase.rule*, que será distribuído juntamente com o aplicativo, ou então através de *download* de um servidor *web*, e será acionado no momento de execução do motor de inferência. O conhecimento é representado nessa base através de regras de produção específicas, personalizadas para o usuário.

Cada regra de produção, contida na base de regras específicas para um dado paciente, é composta de uma precondição sensitiva (uma declaração do tipo “Se”) e uma ação (uma declaração do tipo “Então”). Isto é, cada regra é uma sentença lógica do tipo “Se <condição(ões)> então <ação(ões)>”. Quando uma precondição coincide com o estado atual do conhecimento (a memória de trabalho), o motor de inferência executa a ação relativa àquela regra.

Para representar as regras de produção na base de conhecimentos do *MonitorPrevIntel*, foi especificada uma linguagem própria, que facilitasse o trabalho de formalização do conhecimento utilizado por este trabalho e tornasse essa base independente do código fonte do sistema. A gramática utilizada por esta linguagem utiliza poucos e significativos elementos facilitando sua compreensão e sua manutenção. A Figura 5.2 apresenta uma base de conhecimentos especificada através desta linguagem.

```
ruleBase teste

rule quarta:
  conditions:
    SBPVarianceSleeping = BAIXO;
  actions:
    alert;

rule quinta:
  conditions:
    SBPVarianceSleeping = NORMAL;
  actions:
    normal;

rule sexta:
  conditions:
    heartRate = ALTO;
  actions:
    alert;

.
```

Figura 5.2 - Base de conhecimentos especificada na linguagem definida neste trabalho.

A gramática da linguagem é constituída de palavras reservadas, sinais de pontuação e operadores de comparação. As palavras reservadas são utilizadas para fins específicos na linguagem descritiva, tal como, *conditions* que é utilizada para marcar o início das precondições sensitivas de uma regra. Os sinais de pontuação são utilizados como marcadores no texto, de finalização do arquivo (.), de finalização de precondição ou ação (;) e de início de bloco (:). E os sinais de comparação são utilizados como operadores das precondições

sensitivas. Na Tabela 5.1 são apresentados todos os elementos que fazem parte da linguagem de especificação da base de conhecimentos.

Tabela 5.1 - Elementos da linguagem de especificação da base de conhecimentos.

Tipo	Dicionário da linguagem	Para que são usados
Palavras reservadas	ruleBase	Nome da base de conhecimentos
	rule	Nome da regra de produção
	conditions	Marcador das condições
	actions	Marcador das ações
	normal	Tipos de situações de saúde detectadas
	alert	
	emergency	
Identificadores	Name	Identificador de nome
	Age	Identificador de idade
	Gender	Identificador de sexo
	Weight	Identificador do peso
	VitalSigns	Identificador dos sinais vitais
	SBP	Identificador de Pressão Arterial Sistólica (PAS)
	DBP	Identificador de Pressão Arterial Diastólica (PAD)
	heartRate	Identificador de frequência cardíaca
	respiratoryRate	Identificador de frequência respiratória
	temperature	Identificador de temperatura corporal
	oximetry	Identificador de oxigenação no sangue
	SBPVarianceSleeping	Identificador de variação de PAS durante o sono
	SBPVarianceHomeActivity	Identificador de variação de PAS durante atividades domésticas
	SBPVarianceStand	Identificador de variação de PAS durante o repouso
	MUITO_BAIXO	Identificadores de classificação da variação do sinais vitais
	BAIXO	
	BAIXA	
	NORMAL	
	ALTA	
	ALTO	
	MUITO_ALTO	
Sinais de Pontuação	:	Marcador de início de condições da regra de produção
	;	Marcador de final de linha de condição ou ação
	.	Marcador de final de base de regras
Operadores de Comparação	< <= > <= = #	Símbolos utilizados para fazer as operações de comparação

5.2.2 Classes do MonitorPrevIntel

As classes que compõem o *MonitorPrevIntel* foram organizadas em pacotes de acordo com suas funcionalidades, possibilitando uma visão arquitetural mais detalhada (Figura 5.3). O pacote *mpi.patient* reúne as classes de manipulação dos dados do paciente e de sinais vitais. O *mpi.sms* reúne as classes de manipulação de mensagens de texto curtas (SMS). O pacote *mpi.utils* traz as classes com requisitos não-funcionais do sistema, sendo dividido ainda em um pacote de nível hierárquico inferior (*mpi.utils.compiler*) que contém as classes de manipulação de arquivo e compilação da base de regras de produção. O pacote *mpi.resources* armazena os recursos multimídia utilizados na execução do sistema, que são arquivos de áudio (no pacote *mpi.resources.sounds*), o arquivo contendo os dados pessoais do usuário (no próprio *mpi.resources*) e a base de regras de produção (no pacote *mpi.resources.rules*). Por último o pacote *mpi* que é o de maior nível hierárquico reúne todos os demais pacotes e a classe principal do aplicativo.

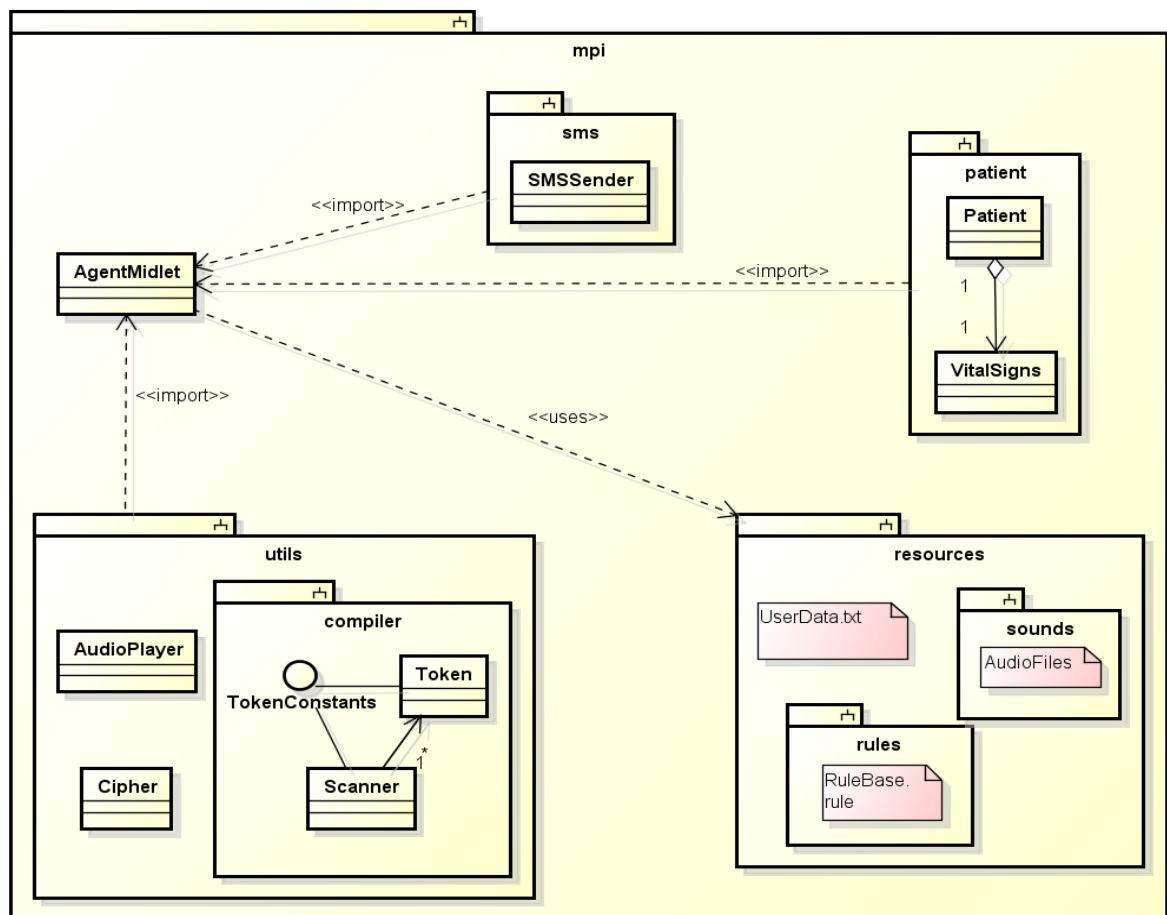


Figura 5.3 - Diagrama de pacotes do sistema.

Observando o diagrama de pacotes, pode-se dizer que o MR (apresentado na Figura 5.1) é composto pelo pacote *mpi.sms* e pela classe *Cipher* do pacote *mpi.utils*. E então todas as demais classes compõem o MI (também presente na Figura 5.1). A Figura 5.4 apresenta o diagrama de classes do sistema, mostrando o relacionamento entre as classes que compõem o *MonitorPrevIntel* e os demais pacotes utilizados como recurso para o sistema, bem como os arquivos utilizados como recurso, quer sejam utilizados (*Audio.wav*, *RuleBase.rule*, *UserData.txt*) ou produzidos (*DataBase*) por ele.

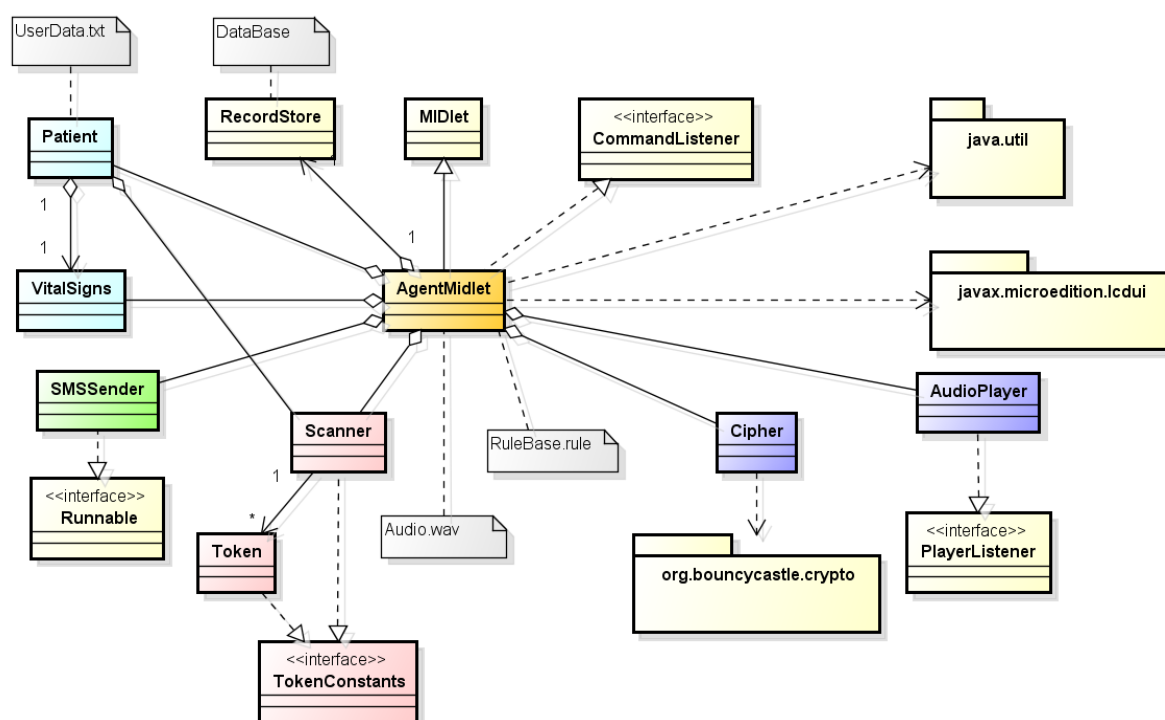


Figura 5.4 - Diagrama de classes do *MonitorPrevIntel*.

No diagrama de classes, cada pacote (ou subpacote) está representado por uma cor específica. Na cor lilás as classes *Cipher* e *AudioPlayer* que fazem parte do pacote *mpi.utils*, na cor rosa as classes *Token* e *Scanner* e a interface *TokenConstants* do pacote *mpi.utils.compiler*, na cor verde a classe *SMSSender* no pacote *mpi.sms*, na cor azul as classes *Paciente* e *VitalSigns* do pacote *mpi.patient*, na cor laranja a classe *AgentMidlet* do pacote *mpi*, na cor amarela temos os recursos, da própria linguagem ou desenvolvidos por outras pessoas, utilizadas pelo sistema e na cor cinza os arquivos utilizados como recursos. A seguir cada classe é descrita individualmente.

A classe *AudioPlayer* implementa a interface *Player* e apresenta mecanismo para manipulação de arquivos de áudio, contendo métodos que permitem a abertura, a execução, pausa, interrupção e fechamento do arquivo. A classe *Cipher* utiliza os recursos da API de

criptografia *Bouncy Castle* *Cripto*, importando as classes *BlockCipher*, *engines.AESLightEngine*, *modes.CBCBlockCipher*, *params.KeyParameter* e *params.ParametersWithIV*, para implementar a cifra de bloco AES-128. A senha é passada ao construtor no momento da inicialização do objeto. Além disso esta classe possui métodos para cifragem e decifragem de mensagens texto (*String*) ou vetores de bytes (*byte[]*), bem como um método que retorna o vetor utilizado na cifra de bloco. Esses métodos são utilizados para criptografar e descriptografar o conteúdo das mensagens SMS.

A interface *TokenConstants* é utilizada para definir as constantes utilizadas como marcadores na leitura dos arquivos. Isto é, define as *tags* utilizadas para classificar os dados lidos dos arquivos, podemos vê-las na Figura 5.5. A classe *Token* implementa esta interface e cria objetos que classificam as palavras ou os dados lidos dos arquivos de acordo com as constantes de *TokenConstants*. Cada objeto desta classe é composto por dois atributos: o *tokenType* (byte) que identifica o tipo do objeto de acordo com a interface e o *lexeme* (*String*) que armazena a palavra, o símbolo ou o número lido. Essa classe define um objeto para a tag *EOF* (fim de arquivo, do inglês: *End Of File*). Além dos métodos de *get()* e *set()*, é definido um método *toString()* que retorna o objeto lido em forma de palavra (*String*), por exemplo, tomando o objeto *Token* que representa o número 10, o método *toString()* para este objeto retorna a *String* “NUMBER: 10”.

```
package mpi.utils.compiler;
public interface TokenConstants {
    public static final byte RULE_BASE = 0;
    public static final byte RULE = 1;
    public static final byte CONDITIONS = 2;
    public static final byte ACTIONS = 3;
    public static final byte IDENTIFIER = 4;
    public static final byte NUMBER = 5;
    public static final byte STRING = 6;
    public static final byte COLON = 10;///  

    public static final byte SEMI_COLON = 11;///  

    public static final byte DOT = 12;///  

    public static final byte WHITE_SPACE = 13;
    public static final byte EOF = 14;
    public static final byte EQUALS = 20;///  

    public static final byte DIFFERENT = 21;///  

    public static final byte GREATER = 22;///  

    public static final byte GREATER_OR_EQUALS = 23;///  

    public static final byte LOWER = 24;///  

    public static final byte LOWER_OR_EQUALS = 25;///  

    public static final byte ERROR = 98;
    public static final byte NONE_ABOVE = 99;
}
```

Figura 5.5 - Interface *TokenConstants*.

A interface *TokenConstants* também é implementada pela classe *Scanner*. Esta é utilizada para leitura dos arquivos contendo as informações pessoais do usuário do sistema e arquivos de sinais vitais e pela transformação destes em objetos *Token*. A classe *Scanner* abre o arquivo, faz a leitura e a identificação dos tipos das palavras, de acordo com as constantes de *TokenConstants* e retorna o *Token* referente a leitura, através do método *nextToken()* e ao final da leitura fecha o arquivo. O método *nextToken()* utiliza o método *readNextChar()* para ler um caractere por vez e, então, junta-os para formar os objetos *Token*. Alguns métodos que o auxiliam no processo de leitura das palavras são:

- *boolean: isIdentifierStart(char)*: identifica se o caractere é válido como início de uma palavra;
- *boolean: isIdentifierPart(char)*: identifica se o caractere é válido como parte de uma palavra;
- *boolean: isNumber(char)*: identifica se o caractere é um número;
- *Token: reservedOrIdOrString(String)*: método que, auxiliado pelos métodos *int: checkIdentifier(String)* e *int: checkReserved(String)*, retorna o objeto adequado a palavra passada como argumento.

Para fazer a classificação dos *Tokens* é necessário que a classe *Scanner* conheça os símbolos utilizados pela linguagem na qual foi escrita a base de conhecimento e as suas regras de produção e o arquivo de dados pessoais do usuário. Por isso a necessidade desses métodos enumerados acima. O método *checkReserved(String)* verifica se a palavra passada como argumento está entre os tipos (*byte[] typeReservedWords*) definidos no vetor de palavras reservadas (*String[] reservedWords*) e retorna o número equivalente ao tipo correto, ou caso não seja uma palavra reservada retorna -1. O método *checkIdentifier(String)* verifica no vetor de identificadores (*String[] identifiersWords*) se a palavra passada como argumento é do tipo *TokenConstants.IDENTIFIER*, caso seja, seu valor é retornado, caso contrário, retorna o valor -1.

A classe *Scanner* faz a separação do arquivo *RuleBase.rule* em *Tokens* que serão interpretados pelo motor de inferência, bem como a compilação do arquivo *UserData.txt* para atributos do objeto *Patient*. É interessante destacar que o método *nextToken()* identifica os espaços em branco, as quebras de linha e as tabulações e os descarta, retornando apenas *Tokens* com conteúdo útil. Outro destaque interessante é que essa classe permite que as demais, que fazem uso dela, interpretem os símbolos à medida que o arquivo é lido, sem que seja necessário fazer sua leitura mais de uma vez ou que seja preciso armazenar um vetor de

Tokens. Desta maneira, evita-se a sobrecarga de processamento e de memória, uma vez que só é necessário manter uma conexão com arquivo aberta.

A classe *Patient* carrega todas as informações pessoais do paciente: nome (*name*), idade (*age*), peso (*weight*), sexo (*gender*) e sinais vitais normais (*normalSigns*). O objeto desta classe pode ser criado através de um construtor que tenha como argumento um arquivo e a partir deste se inicializa os atributos do *Patient*, ou de outro construtor que receba os próprios atributos como argumento. Os seus métodos são simples *get()* e *set()*. O atributo *normalSigns* é um objeto da classe *VitalSigns* que contém os parâmetros normais de medidas fisiológicas para o usuário.

A classe *VitalSigns* define objetos compostos por dados de sinais vitais, que são: frequência cardíaca (*heartRate*), pressão sanguínea sistólica (*systolicBloodPressure*) e diastólica (*diastolicBloodPressure*), frequência respiratória (*respiratoryRate*), oxigenação do sangue (*bloodOximetry*) e temperatura (*temperature*). Estes dados fisiológicos particulares foram escolhidos devido a sua importância no acompanhamento de saúde de pessoas com doenças cardiovasculares. Essa classe contém, além dos métodos *get()* e *set()*, um método *toString()* que transforma os seu atributos em uma *String* com a hora em que foram convertidos.

SMSSender envia as mensagens de texto, relatando informações sobre o usuário do sistema. Essa classe estabelece uma conexão com o destinatário da mensagem, envia seu conteúdo e encerra a conexão. Um objeto dessa classe tem como atributo um número de telefone do destinatário, o número da porta a ser utilizada para o envio e o conteúdo da mensagem. Além disso, a classe possui um *flag* booleano (*isSent*) utilizado para verificar se o envio da mensagem foi realizado. Os únicos métodos que esta classe tem são os métodos *run()* e *send()*. O primeiro método é a implementação do método abstrato da interface *Runnable*, já que *SMSSender* a implementa. É nele que ocorre todo o funcionamento descrito no início desse parágrafo, caso haja algum problema nesse processo e a mensagem não seja enviada, o *flag* da classe é marcado como *false*. Já o método *send()* é utilizado para executar o método *run()* e retorna o valor do booleano *isSent*.

A principal classe do sistema é a *AgentMidlet*. É ela que reúne todas as funcionalidades das demais classes para executar o fluxo de ações do *MonitorPrevIntel*. A Figura 5.6 apresenta o caso de uso do *software*. Ela é a classe que será executada quando da inicialização do sistema, por herdar da classe *MIDlet*. Essa classe implementa a classe *CommandListener* sobrescrevendo o método *commandAction(Command, Displayable)* para controlar as interações com as ações do usuário. A classe *AgentMidlet* é responsável por

gerenciar todas as atividades do sistema. Nela é implementado o mecanismo do motor de inferência e, por isso, podemos chamá-la de agente inteligente, ou, simplesmente, agente. Além disso, a classe gerencia a manipulação da interface do usuário, a montagem e criptografia de mensagens a serem reportadas através de SMS e outras funções de menor relevância.

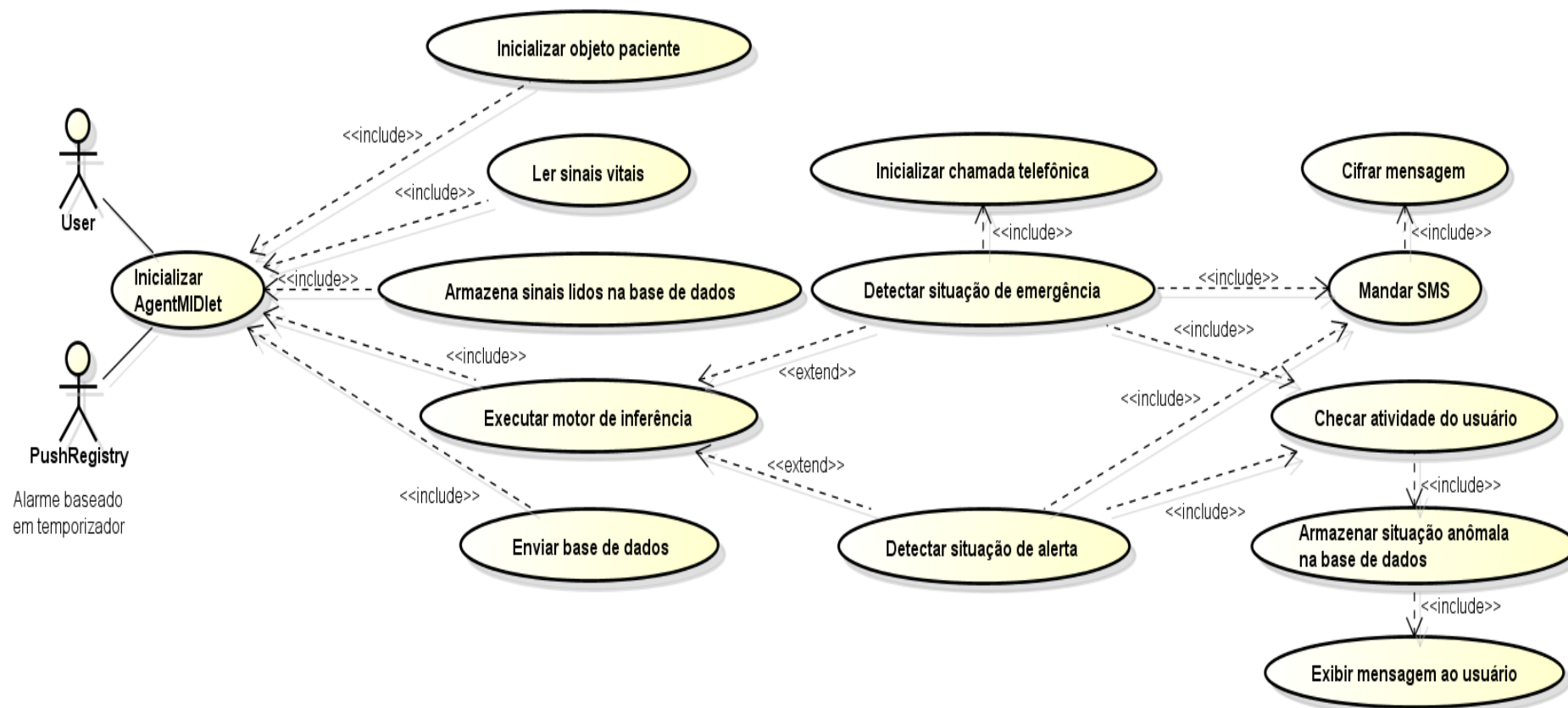


Figura 5.6 - Caso de uso do *MonitorPrevIntel*.

Ao ser inicializada, a classe *AgentMidlet* carrega os dados pessoais do usuário do sistema inicializando um objeto *Patient*. Ao ser acionada pelo AMS, ela carrega uma interface gráfica para o usuário onde irá expor as atividades que estão sendo executadas. Após isso, carrega os dados atuais dos sinais vitais em um objeto *VitalSigns* e os armazena fazendo uma chamada ao método *storeData(String)*. Esse método abre ou cria, caso ainda não exista, um *RecordStore* que servirá como base de dados (chamado de *DataBase*) do aplicativo, grava os dados recebidos como parâmetro, fecha o arquivo e retorna o número da linha na qual foram armazenados os dados.

Após isso, os dados são entregues ao motor de inferência. O sistema de produção foi implementado no método *fireRuleBase*, que recebe como parâmetros o nome e local da base de regras (*baseName: String*) e os dados de sinais vitais (*vS: VitalSigns*) carregados, que formam a memória de trabalho. Esse método irá interpretar *vS* a partir das regras contidas na *baseName* (que é o arquivo *RuleBase.rule*) e ativará, ou não, situações de normalidade, alertas ou emergências de saúde. O diagrama de atividades do *fireRuleBase* pode ser observado na Figura 5.7.

Por falta de uma equipe multidisciplinar que auxiliasse a criar mecanismos e procedimentos de ação mais detalhados, foi decidido que as ações realizadas em cada situação seriam as seguintes:

- Normal: as medidas e o nome da regra que dispararam a situação, a data e hora da detecção da mesma são armazenadas numa base de dados e ainda ajusta o intervalo de execução automática do aplicativo para 120 segundos;
- Alerta: as medidas e o nome da regra que dispararam a situação, a data e hora da detecção da mesma são armazenadas numa base de dados. É reportada uma mensagem SMS com o mesmo conteúdo, além disso, uma mensagem informando a alteração do sinal vital do paciente é informada na tela do celular e o intervalo de execução automática do aplicativo é reduzido para 60 segundos;
- Emergência: as medidas e o nome da regra que dispararam a situação, a data e hora da detecção da mesma são armazenadas numa base de dados. É reportada uma mensagem SMS com o mesmo conteúdo, e ainda, uma mensagem informando a alteração do sinal vital do paciente é informada na tela do celular direcionando-o para a inicialização de uma chamada para uma central de emergência médica e o intervalo de execução automática do aplicativo é reduzido para 30 segundos.

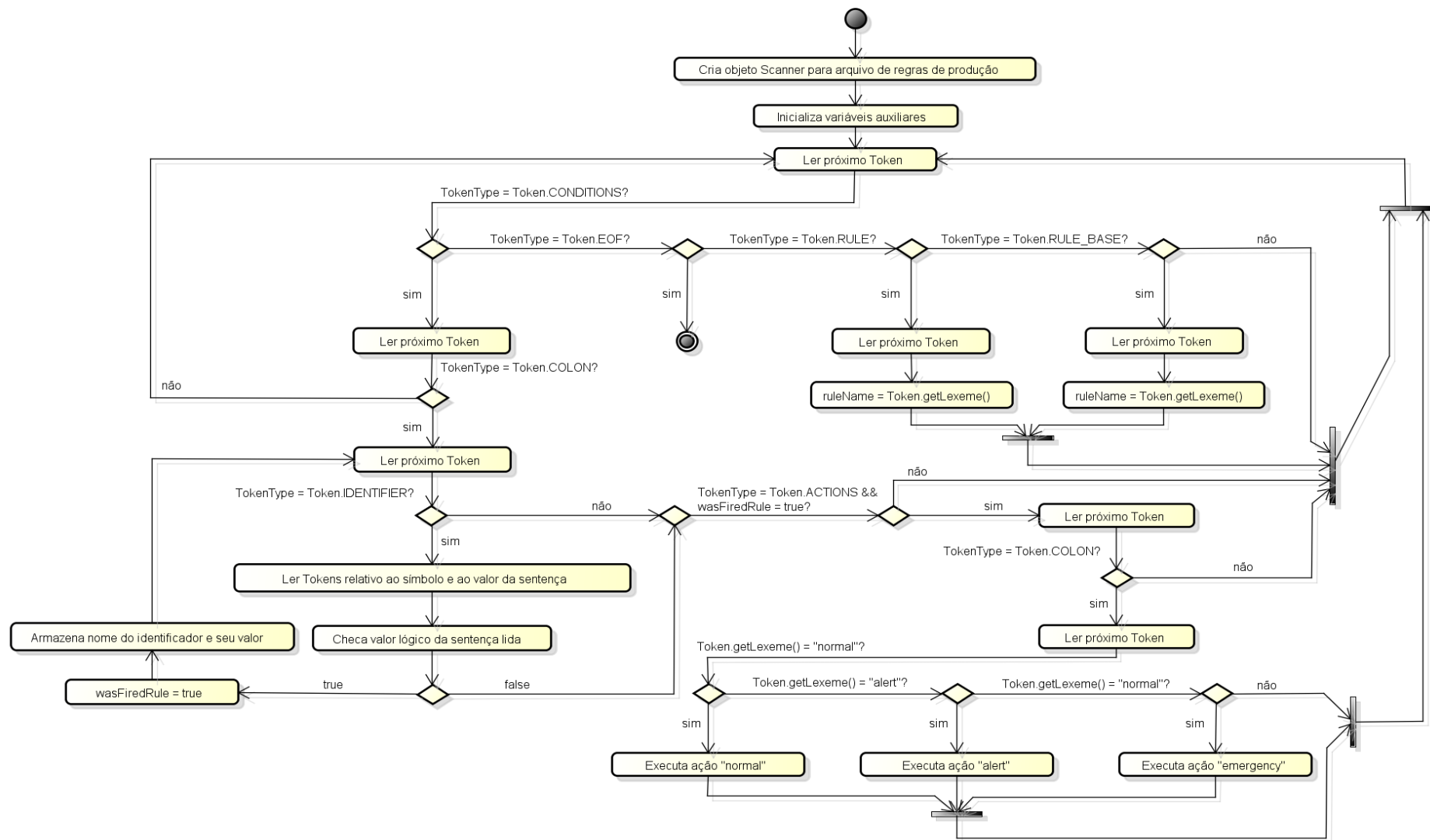


Figura 5.7 - Diagrama de atividades do método *fireRuleBase()* da classe *AgentMidlet*.

O método *fireRuleBase()* detecta a ação e aciona o método *action()* para executá-lo. Esse método recebe como parâmetro um byte indicando o tipo da ação (0 – normal, 1 – alerta, 2 – emergência), o nome da regra que o acionou e um número telefônico para onde será enviada a mensagem informando os dados que ativaram a regra. A Figura 5.8 apresenta o diagrama de atividades do método *action()*.

Para enviar a mensagem SMS, o método *action()* faz uma chamada ao método *sendSMS()* que recebe como parâmetro o telefone do destino (*destinationAddress: String*) e o conteúdo da mensagem (*content: String*) a ser enviada. Nesse método é criado um objeto da classe *SMSSender*. A mensagem é quebrada em pacotes de 160 caracteres¹⁷ e então cada pacote é enviado para o destinatário. Caso a mensagem seja enviada, a mensagem “*Message sent to <destinationAddress>.*” é exibida na interface gráfica do celular.

Após encerrar a interpretação dos dados fisiológicos, segundo a base de conhecimento, a classe executará o método *reportHistoric()*. Este método é responsável por reportar as informações do paciente armazenados na base de dados para algum destino através do método *sendSMS()*. Os dados são enviados a cada três horas ou quando o tamanho da base ultrapassa os 600 bytes. Caso os dados sejam enviados com sucesso através de SMS a base de dados é apagada para gerar espaço para novos dados. Caso contrário, a base de dados é mantida com os dados atuais.

Por último, é feita uma chamada ao método *activeTimer()* que é responsável por agendar a execução automática do sistema de acordo com o tempo determinado pelas execução das ações das regras de produção. Esse método programa a execução automática usando chamadas ao método *registerAlarm* da classe *PushRegistry* do perfil MIDP 2.0. Isso permite que o programa seja ativado sem a intervenção do usuário.

¹⁷ Tamanho da mensagem escrita no formato 7-bits GSM, que é padrão obrigatório a todos os dispositivos de telefonia móvel.

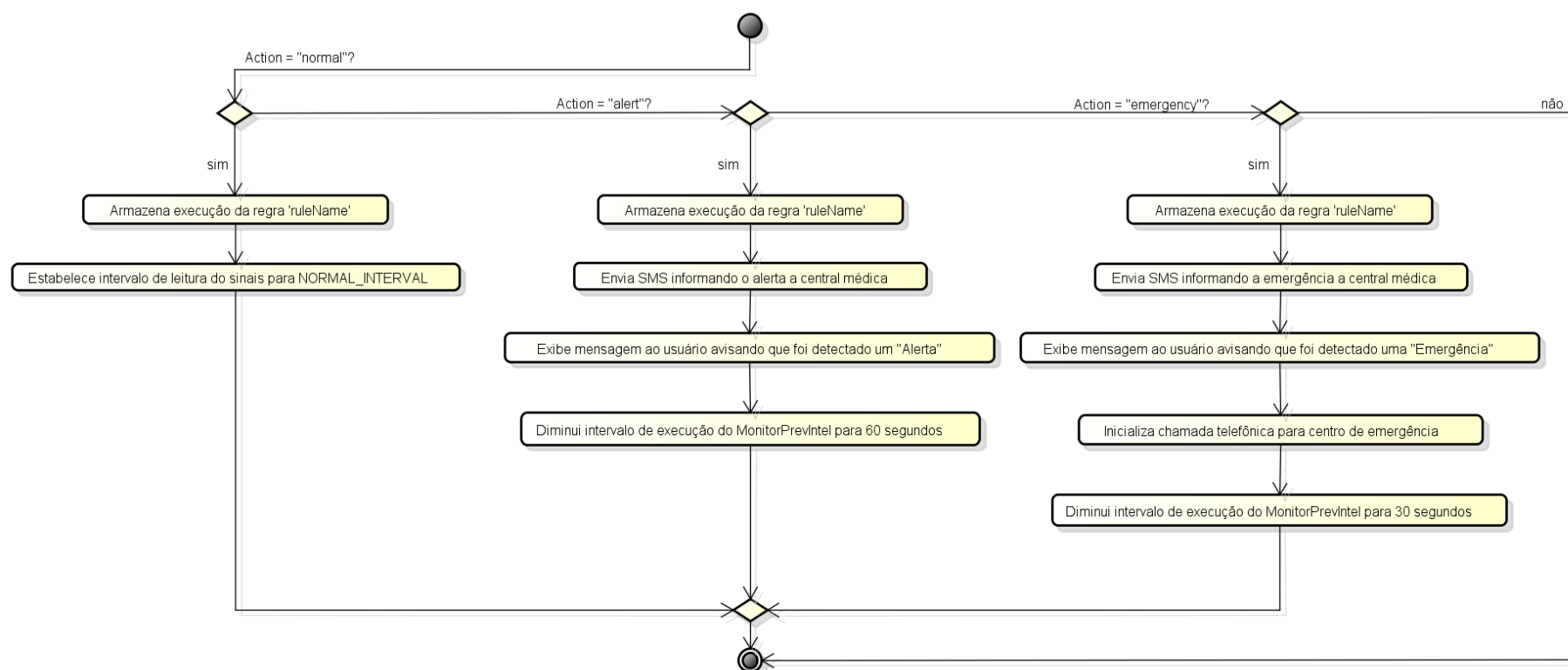


Figura 5.8 - Diagrama de atividades do método `action()` da classe `AgentMidlet`.

5.2.3 Base de Dados

A base de dados do aplicativo é o local onde ficam armazenados os dados fisiológicos do paciente. Cada leitura dos sinais vitais realizada será gravada. Esses dados serão utilizados na construção do histórico de saúde do paciente. Além disso, percebeu-se a necessidade de registro das situações de anormalidade detectadas pelo motor de inferência, para que essas pudessem ser avaliadas dentro do contexto em que foram aferidas. Por exemplo, as características que levaram a detecção de uma situação de emergência logo após uma emissão de situação de alerta.

As informações são armazenadas de maneira a ocupar o mínimo de espaço em memória e são relatadas frequentemente para uma central de supervisão médica. Como o JME não tem um sistema de banco de dados, foi utilizado o *Record Management System* (RMS) que realiza a gravação de dados, persistindo-os, em arquivos identificados por objetos *Record Store*. Cada arquivo deste é um vetor de *Records* que é uma linha do arquivo que armazena os dados recebidos, identificados por números que facilitam sua manipulação.

5.3 *Funcionamento*

Após a descrição das características e dos componentes do *Monitor Intelligent Prevention*, nesta seção é apresentado o funcionamento do sistema. O aplicativo deve ser colocado no telefone móvel com suporte a Java a partir de uma instalação direta no dispositivo ou de uma instalação *Over The Air* (OTA)¹⁸. Ao ser acionado, o MIDlet inicializa os dados do usuário em um objeto *Patient* (*user*) e os dados fisiológicos, recebidos da medição, em um objeto *VitalSigns* (*vS*), armazenando o conteúdo desse, a hora e data atual na base de dados e, então, o entrega ao motor de inferência.

¹⁸ OTA é um padrão para transmissão e recepção de informações relativa a aplicações que utilizam sistemas de comunicação sem fio, muito utilizada para instalação de aplicações ou carregamento de dados em telefones móveis.

Ao acionar o sistema de produção, a base de conhecimentos é carregada do arquivo *RuleBase.rule* e os dados recebidos são carregados para a memória de trabalho. O motor de inferência então irá interpretar a memória de trabalho de acordo com a base de regras e a partir do teste das condicionais das regras irá ativar ações de emergência, alerta ou normalidade. Na medida em que a base de conhecimentos é lida, os dados são interpretados e as regras que forem atendidas são executadas. Assim, o agente pode interromper a interpretação para ativar uma ação e, depois, voltar para o fluxo normal de execução, para a interpretação do conhecimento pelo motor de inferência.

Como, ao longo do trabalho, não houve a montagem de uma equipe multidisciplinar, que incluísse profissionais da área médica para auxiliar na criação de ações mais específicas para o agente, foi decidido utilizar um sistema de detecção de situações anormais de saúde com base nas três divisões utilizadas por COPETTI (2010). Com essa divisão, que resulta em ações normais, de alerta ou emergenciais, o sistema é capaz de analisar se as condições de saúde do paciente estão muito, pouco ou não alteradas.

Assim, quando uma situação normal é ativada o sistema armazena a informação de que uma determinada regra foi acionada. Por exemplo, a informação *“The rule ‘primeira’ triggered an normal status.”* é armazenada quando a regra por nome *primeira* for acionada. Quando o motor de inferência ativa uma ação de alerta, o sistema armazena a informação de que aquela regra foi acionada e envia uma mensagem SMS de alerta contendo os dados do paciente que ativaram a regra para o centro de supervisão médica. O modelo do conteúdo de um SMS utilizado para carregar uma mensagem de alerta ou de emergência é apresentado na Figura 5.9 (a). Na Figura 5.9 (b) é exibido um exemplo de mensagem de emergência.

- (a) *<nomeDoUsuário> triggered an <alert >|<emergency>.
<vitalSign> is <MUITO_BAIXO>|<BAIXO, ALTO>|<MUITO_ALTO>.*
- (b) *Maria Joaquina triggered an emergency.
SBP Variance Sleeping is MUITO_ALTO.*

Figura 5.9 - Modelo (a) e exemplo (b) de SMS enviada pelo aplicativo informando uma situação de anormalidade no sinais vitais.

Na situação de alerta o sistema ainda envia uma mensagem, através da interface gráfica do telefone móvel, ao usuário informando um alarme e dizendo qual sinal vital foi detectado como alterado e, por último, o sistema reduz o tempo do agendamento da próxima execução. Ao detectar uma ação de emergência, o sistema age praticamente da mesma maneira que na situação de alerta. A diferença está na diminuição do tempo de agendamento da próxima execução do aplicativo e na mensagem apresentada através da interface gráfica do

celular na qual é apresentado um texto maior sugerindo que o usuário realize uma chamada para uma central de pronto atendimento e um botão que já inicializa a chamada.

Durante a interpretação das regras o agente pode ainda interromper sua execução para questionar o usuário sobre a sua atividade física no momento, o qual pode selecionar uma das seguintes atividades: *Stand*, *Home Activity*, *Sleeping*, *Other*. Caso o sistema detecte que já passa das 22h (vinte e duas horas) e ainda não é mais que 6h (seis horas) o agente admite que o usuário está em seu horário de sono e não o questiona sobre sua atividade. O sistema foi implementado para considerar essas quatro atividades físicas específicas porque elas são utilizadas pelas regras de produção apresentadas em COPETTI (2010) para definir a ação a ser realizada pelo agente. Por exemplo, o usuário tem um aumento no valor da pressão arterial sistólica em 21mmHg, caso sua atividade seja *Sleeping* esse sinal vital pode ativar uma ação de emergência, caso sua atividade seja *Home Activity* esse sinal vital pode ativar uma ação de alerta. No Apêndice A é descrito o processo de classificação da variação da pressão arterial sistólica e da frequência cardíaca do usuário de acordo com a sua atividade física, conforme apresentado na Tese de Doutorado de COPETTI (2010).

Ao finalizar a interpretação das regras da base de conhecimentos, o aplicativo faz uma chamada ao método *reportHistoric()* que irá reportar a base de dados para a central de supervisão médica através de mensagens SMS. Este relatório será enviado em horários predefinidos ao longo do dia, às 7h, 10h, 13h, 16h, 19h e às 22h, ou então quando verificar que a base de dados tem tamanho superior ou igual a 600 bytes.

Por último o sistema aciona o agendamento da execução para o tempo definido pelas ações usando os recursos da classe *PushRegistry*. Com isso a ativação do MIDlet é realizada independentemente da ação do usuário. É importante destacar que a ativação de maneira automática não impede que o usuário execute o *MonitorPrevIntel* de maneira manual. Ainda sobre a interação com o usuário vale destacar que toda vez que o sistema irá interagir com ele, ao exibir alguma mensagem ou algum questionamento, são emitidos sinais sonoros e é acionado o vibrador do celular para que o usuário perceba que sua atenção é requisitada pelo aplicativo móvel.

Capítulo

6

Testes e Resultados

“Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção.”

Steve Jobs

Para verificar a consistência do aplicativo *MonitorPrevIntel* em relação aos objetivos propostos e ao comportamento e a aplicabilidade do mesmo diante de diferentes situação, o MIDlet foi implementado e posto em emulação para realização de alguns casos de teste. Durante a construção do aplicativo testou-se os componentes à medida que eram desenvolvidos, como proposto pelas atividades do modelo de desenvolvimento de *software*. Ao final de cada ciclo do modelo espiral, um dos componentes do sistema passava por um processo de teste de caixa preta e validação. Para teste dos códigos criados foi utilizado o emulador Wireless Toolkit 2.5.2 e o emulador da própria IDE NetBeans 6.9.1.

No primeiro ciclo foi desenvolvido a classe de SMS, a *SMSSender*. Para testá-la foi desenvolvido um MIDlet só de envio de mensagens, o *SMSMidlet*. O envio das mensagens através desse MIDlet foi testado com o uso da *Wireless Messaging API* (WMA) da ferramenta Wireless Toolkit 2.5.2. A interface do WMA funciona como a tela de um celular que exibe mensagens recebidas ou ainda envia mensagens para um destino através de uma porta específica. No ciclo de desenvolvimento da classe *Cipher* e para testar o envio de SMS criptografada, mais uma vez foi utilizado a classe *SMSMidlet*. A Figura 6.1 apresenta a interface gráfica do WMA ao receber a mensagem “*SBP Variance Sleeping is MUITO_ALTO*” criptografada com a senha “*ManuellaDCSilva.*”.

No ciclo seguinte foram desenvolvidos os elementos de interface gráfica do aplicativo. Para teste destes componentes foi desenvolvido o *UIMidlet*. Com este MIDlet foram testadas, no emulador Wireless Toolkit 2.5.2, as interações entre as telas e também o uso da classe *AudioPlayer*. Isto porque esta classe é utilizada como auxílio à interface gráfica, pois uma vez

que o programa executa sem a intervenção do usuário seria muito difícil ele perceber uma mensagem exibida na tela do celular.

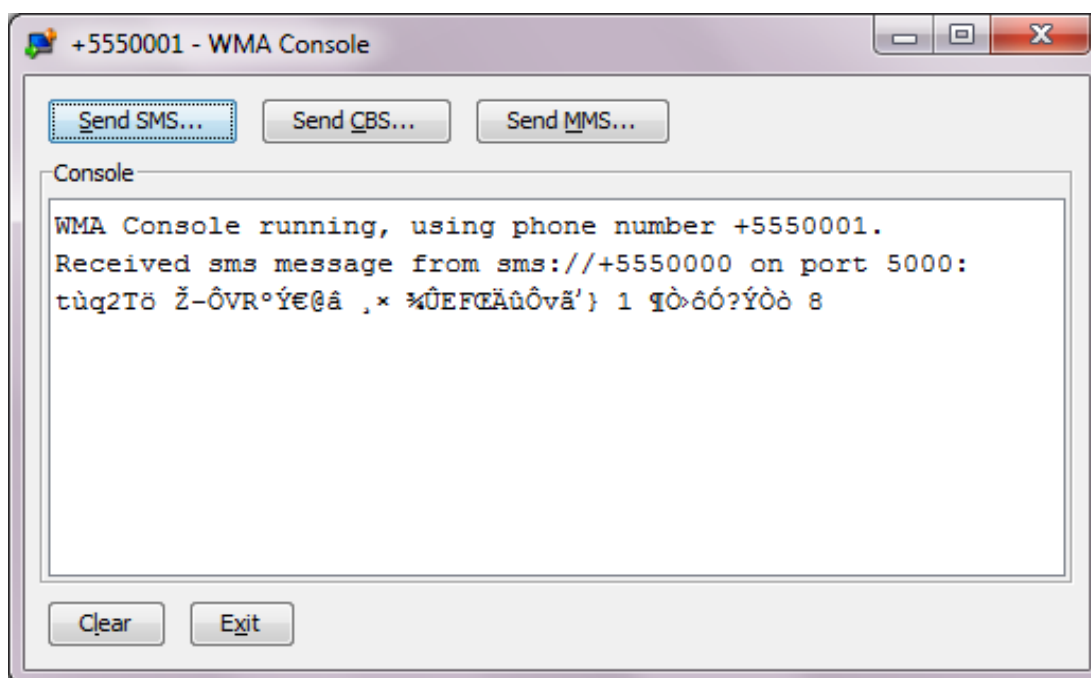


Figura 6.1 - Interface gráfica do WMA da ferramenta Wireless Toolkit 2.5.2.

Em outro ciclo do modelo espiral foram modificadas as classes *Scanner*, *Token* e *TokenConstants* do programa JEOPS¹⁹, Copyright 2000 Carlos Filgueira Filho, para atender a especificação da linguagem criada para modelar a base de conhecimentos do aplicativo *MonitorPrevIntel*. Então o teste dessas classes foi realizado para validação da linguagem de descrição das regras de produção. Estas classes foram testadas através de métodos desenvolvidos especificamente para elas dentro do *SMSMidlet*. Nesses testes já foi feito a integração entre leitura de arquivo e envio de algumas mensagens SMS criptografadas. Os ciclos seguintes serviram para implementação do MIDlet com inteligência, o *AgentMidlet*.

Antes de partir para descrição da verificação do *AgentMidlet*, é necessário montar seu ambiente de testes. Este deve englobar os recursos necessários que possibilitem a análise do comportamento do agente em diferentes situações. Para isso, é necessária a construção de pelo menos uma base de conhecimento formada por um conjunto de diferentes regras de produção, a geração de dados sintéticos os quais representem sinais vitais, permitindo a criação de diferentes cenários de investigação, e uma ferramenta que possa acompanhar o fluxo de atividades do aplicativo.

¹⁹ Programa distribuído sob os termos da *GNU Lesser General Public License*. Disponível no endereço: <<http://www.cin.ufpe.br/~jeops/>>.

Para construção da base de conhecimentos foi utilizado o conjunto de dados e de regras de produção apresentado na Tese de Doutorado de Copetti (2010). Esta tese traz 63 regras destinadas ao monitoramento de pessoas com doenças cardiovasculares, com base na atividade do paciente, pressão arterial sistólica e diastólica, frequência cardíaca e temperatura ambiente. As regras são escritas na forma *if <precondition(s)> then <action(s)> (prioridade da regra²⁰)*, a Figura 6.2 traz um exemplo.

if (atividade is dormir) and (freqCard is alta) then (acao is alerta) (0.7)

Figura 6.2 - Exemplo de regra de produção do formato apresentado em Copetti (2010).

Do conjunto de regras desenvolvido em Copetti (2010), apenas 21 regras (Anexo A) de produção puderam ser utilizadas para construir a base de conhecimentos do *MonitorPrevIntel*, pois ao longo da Tese só havia indicação de como classificar o desvio de pressão arterial sistólica, durante a execução de diferentes atividades, e a medida da frequência cardíaca. As demais regras, 42 no total, utilizavam alguma outra medida que não tinha classificação explícita no trabalho.

A geração de dados, chamados dados sintéticos, é importante para que sejam criados os mais diferentes cenários para validação do monitoramento inteligente de saúde. Os dados sintéticos utilizados para alimentar o sistema de produção foram gerados de maneira aleatória, por não ser encontrada nenhuma base de dados de referência disponível com os sinais vitais utilizados para monitoramento neste trabalho.

Para o acompanhamento da sequência das atividades do aplicativo inteligente de monitoramento, como já mencionado, foi utilizada a ferramenta de emulação Wireless Toolkit 2.5.2. Esse emulador possui um conjunto de funcionalidades que permitem o acompanhamento da execução do MIDlet e a visualização da saída padrão. A saída padrão é utilizada para exibir a sequência de atividades de maneira mais detalhada, exibindo as mensagens de retorno dos métodos e possibilitando um *debug* manual do programa.

Para iniciar os testes do *MonitorPrevIntel*, faz-se necessário seguir os seguintes passos:

1. Abrir o emulador WTK;
2. Criar projeto com base no arquivo JAD (*File > Creat Project from JAD/JAR file...*);

²⁰ Este campo é um peso, que vale de 0 (menos significativo) a 1 (mais significativo), para indicar a importância da regra de produção.

3. Executar o projeto através do OTA (*Project > Run via OTA*). Ao selecionar a opção *Run via OTA*, aparece uma tela para seguir os passos de instalação do aplicativo. Ao final da instalação o aplicativo pode ser executado;
4. Para acompanhar o envio das mensagens de texto é recomendado abrir o WMA (*File > Utilities > WMA Console*) que permite visualizar as mensagens SMS.

Um fator importante desta etapa de testes é a possibilidade de identificar os erros de execução do aplicativo. Um dos erros localizados foi um problema de controle de acesso ao servidor das mensagens SMS. O erro acontecia quando o sistema enviava várias mensagens em sequência a uma conexão aberta para envio, que ainda era utilizada, e uma nova conexão já era realizada pela mensagem seguinte. Para solucionar este problema, foi utilizado uma *thread* que interrompe o método de envio da classe *AgentMidlet* por 20 milissegundos. Esta modificação garante tempo suficiente para que o servidor envie a mensagem SMS e feche a conexão, e a próxima mensagem faça uma nova conexão com o servidor sem problemas de acesso a ele.

Uma modificação também proveniente dos testes foi a utilização de um mecanismo que permitisse a exibição das mensagens na tela para o usuário, que ficassem disponíveis durante um intervalo de tempo determinado ou enquanto o usuário não acionasse o comando para sair da aplicação. Este mecanismo foi implementado ao final do código do método *startApp()* do MIDlet, para que fosse a última ação a ser executada.

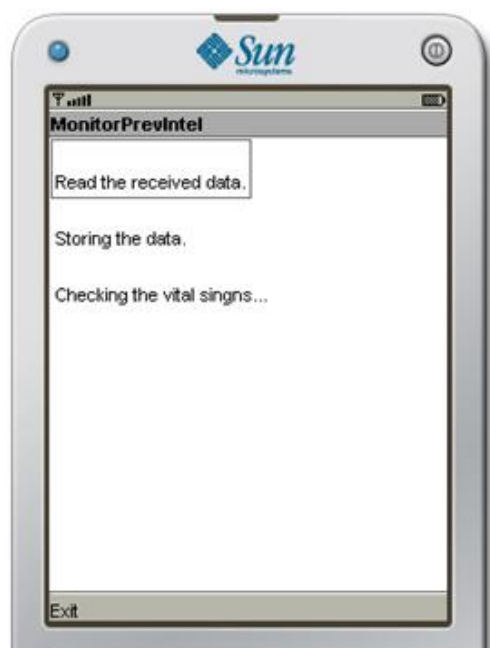
Ao final dos testes temos como resultado o aplicativo para telefones móveis *MonitorPrevIntel* que monitora sinais vitais de pessoas com doenças cardiovasculares e emite alertas sobre a situação de saúde dos mesmos através de um processo ubíquo e inteligente. Para exemplificar a utilização do sistema foi elaborado um cenário de uso que serviu como caso de teste base para a apresentação das telas de interface gráfica do agente móvel de monitoramento, mostradas nas Figura 6.3, Figura 6.4 e Figura 6.5.

Esse caso de teste toma por base uma usuária fictícia de 25 anos que pesa 60kg e tem sinais vitais normais com os valores: 100bpm de frequência cardíaca, 18ipm de frequência respiratória, 120/80mmHg de pressão arterial, 97,5% de oxigenação do sangue e 36°C de temperatura corporal. Utilizando como base de conhecimento o arquivo exposto no Apêndice A e como sinais vitais de entrada os dados sintéticos apresentados na Tabela 6.1 para representar oito medições de sinais vitais dessa usuária.

Tabela 6.1 - Dados sintéticos que representam oito medições.

	Frequência Cardíaca	Frequência Respiratória	Pressão Arterial Sistólica	Pressão Arterial Diastólica	Oxigenação no Sangue	Temperatura Corporal
Medição 1	100	16	120	80	100	37
Medição 2	90	15	141	90	95	38
Medição 3	160	30	120	75	98	37
Medição 4	100	17	120	80	95	38
Medição 5	90	16	110	60	90	36
Medição 6	100	20	160	90	97	38
Medição 7	110	19	120	85	99	37
Medição 8	112	18	120	80	95	36

Para simular a leitura dos dados, eles foram colocados em oito arquivos de texto (nomeados *ReceivedData_n.rcv*, onde *n* indica o número da medição e varia de 1 a 8) e a cada execução do agente era realizada a leitura de um desses arquivos simulando a entrada dos dados de sinais vitais da usuária. Ao iniciar a execução, é exibida uma mensagem informando a leitura dos dados, seu armazenamento e outra mensagem indicando o início do processo de verificação dos sinais vitais da usuária através do agente (Figura 6.3 (a)). Caso nenhuma regra da base de conhecimentos seja ativada, como quando a usuária estiver em repouso (*Stand*) para a primeira medição, a próxima mensagem que aparecerá na tela será avisando o intervalo de tempo para a próxima execução (Figura 6.3 (b)).



(a)



(b)

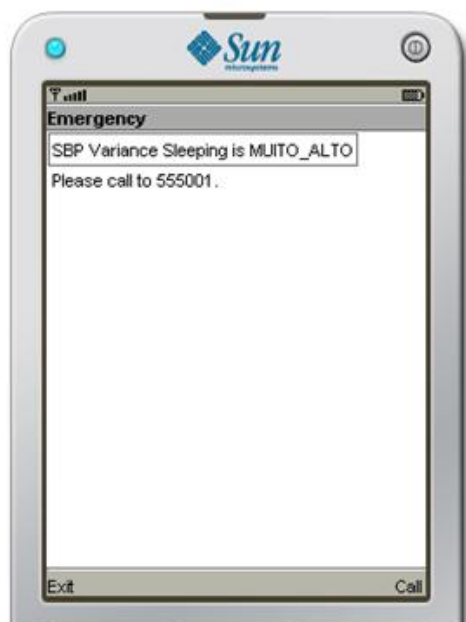
Figura 6.3 - Telas do aplicativo: (a) exibindo as mensagens iniciais, (b) após a execução do aplicativo com a medição 1 da Tabela 6 e o usuário em repouso.

Quando o sistema verifica o valor verdadeiro em algum dos condicionais ele questiona ao usuário sobre sua atividade (Figura 6.4 (a)) e de acordo com sua escolha é ativada uma ação. Quando lida a medição 2 e o sistema detectar que está em horário de repouso do usuário, a atividade será registrada como *Sleeping* e será acionada a regra de produção *segunda*, que ativa uma ação de emergência como consequência da variação extremamente alta da pressão arterial sistólica durante o sono. Então começará a tocar um sinal sonoro, o aparelho móvel

vibrará e o aplicativo exibirá uma mensagem informando ao paciente a variação do seu sinal vital e a sugestão de ligar para a central de pronto atendimento (555001), através do botão *Call* (Figura 6.4 (b)).



(a)

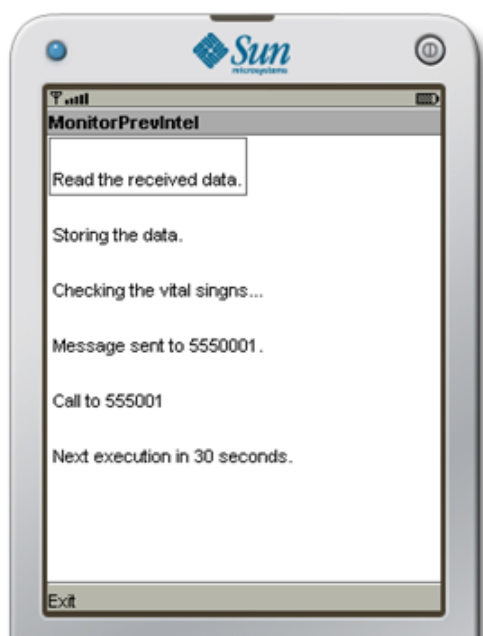


(b)

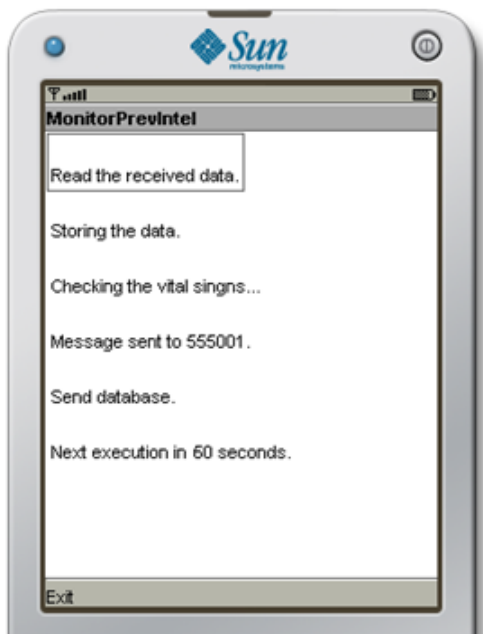
Figura 6.4 - Telas do aplicativo: (a) questionando o usuário sobre sua atividade e (b) exibindo uma mensagem de emergência.

Quando o usuário seleciona o comando do botão *Call* será direcionada uma chamada para o número apresentado e após a chamada o aplicativo continuará sua execução, informando o tempo de intervalo para a próxima execução (Figura 6.5 (a)). Quando uma ação de alerta é ativada, como no caso da medição 8, em que a frequência cardíaca está acima de

99bpm, e a atividade do usuário não é *Sleeping* então aparecerá na tela uma mensagem informando que uma mensagem foi enviada para o número 555001 (que representa a central de médica) e, como a base de dados estava cheia (máximo de 600 bytes), uma mensagem informando o envio de seu backup através de SMS (Figura 6.5 (b)).



(a)



(b)

Figura 6.5 - Telas do aplicativo: (a) exibindo mensagens após execução de ação de emergência, (b) após execução de ação alerta seguida de envio de backup da base de dados.

Durante a execução do caso de teste, simulando oito medições de sinais vitais da usuária fictícia de 25 anos, o sistema se comportou da maneira esperada criando os alarmes e

as interações (avisos sobre emergências e alertas, emissão de sinais sonoros e vibratórios, contato com a usuária e espera pelos seus comandos) previstas pelo fluxo de atividades do *MonitorPrevIntel*. Além desse caso de teste, foram realizados outros dos quais uma amostra foi colocada no Apêndice C – Testes. Em todos os casos apresentados a detecção de situações anormais ou normais nos sinais vitais foi percebida adequadamente, gerando as ações de emergência, alerta ou normalidade esperadas.

Capítulo

7

Considerações Finais

“Tudo tem começo e meio. O fim só existe para quem não percebe o recomeço.”

Luiz Gasparetto

O uso de sistemas de monitoramento de saúde remota mostra-se como uma técnica bastante interessante no auxílio à assistência para pacientes com doenças crônicas, como no caso das pessoas com doenças cardiovasculares, foco escolhido para desenvolvimento deste trabalho. Não só por auxiliar o acompanhamento personalizado do estado fisiológico dos pacientes por especialistas da área médica, mas também por possibilitar a implantação de diversos mecanismos de execução automática, entre esses, o apresentado neste trabalho, a detecção de situações de saúde que fujam da normalidade. Dessa maneira, a contribuição inicial deste trabalho foi apresentar as principais propostas, pesquisas e aplicações na área de tratamento remoto de saúde.

Ao fazer o levantamento no estado da arte, foi possível identificar que a utilização de sistemas pervasivos tem grande impacto na maneira de abordar o telemonitoramento de saúde (COPETTI, 2010). Isso não seria diferente em situações de monitoramento através de sistemas ubíquos, os quais adicionam o parâmetro mobilidade aos sistemas de monitoramento. Com isto, concluímos que a utilização de sistemas ubíquos facilita a adaptação dos usuários aos equipamentos e permite que eles realizem suas atividades cotidianas de maneira comum, sem ficar restritos a determinados espaços físicos, garantindo certo conforto e segurança em suas atividades diárias.

Assim, apresentamos a principal contribuição deste trabalho que é trazer a proposta, o planejamento e a implementação de um sistema de monitoramento remoto utilizando o conceito de ubiquidade em aparelho de baixo custo e uso corriqueiro da população. Ao longo dessa construção foram identificadas as principais tecnologias que podem ser utilizadas para este fim, as características básicas que um sistema como esse deveria apresentar, bem como

quais técnicas implementar e qual seria a principal função do sistema ao monitorar a saúde dos pacientes com doenças cardiovasculares.

O aplicativo para telefones celulares, *MonitorPrevIntel*, apresentado de maneira mais detalhada no capítulo 5, foi desenvolvido para dar suporte ao monitoramento remoto ubíquo de pessoas com doenças cardiovasculares, a partir de um mecanismo de inteligência que utiliza regras de produção para detecção de situações de alerta. Assim, o sistema oferece uma importante ferramenta de auxílio à análise médica, pois permite o monitoramento continuado da saúde do paciente. Em adição, ele reduz o desconforto do paciente que pode manter suas atividades rotineiras praticamente sem interferência, uma vez que o programa é executado no seu aparelho celular e só é percebido quando o usuário é questionado sobre determinadas situações ou avisado de condições críticas de sua saúde.

O aplicativo contém um motor de inferência embutido no *software*, o qual é responsável por interpretar a memória de trabalho de acordo com as regras de produção (Anexo A), personalizadas para cada paciente e estabelecidas por especialistas em saúde. Essa é a parte que traz eficiência ao sistema uma vez que é um componente inteligente que funciona de acordo com as características individuais do paciente permitindo que o MIDlet faça um acompanhamento particularmente adequado ao seu usuário.

O sistema integra componentes para análise e notificação de situações críticas de saúde. A parte de análise inteligente é executada pela máquina de inferência que dispara as demais ações do *software*. A notificação de situações críticas é realizada através do envio de mensagens de texto criptografadas que informam a central de atendimento de saúde as condições dos sinais vitais do paciente que deliberaram aquela notificação. A utilização de mensagens SMS para reportar alertas mostrou-se adequada por ser um mecanismo que está presente em qualquer aparelho de telefonia móvel, desde que coberto por sinal da operadora de telefonia, e também por seu baixo custo²¹.

Ainda sobre o serviço de envio de SMS há muitas críticas quanto à confiabilidade do serviço e à privacidade de seu conteúdo. Para garantir a privacidade dos dados transmitidos através destas mensagens, o sistema utiliza criptografia AES com chave de 128 que é interna ao código do programa. Quanto à confiabilidade, o sistema utiliza um método que retorna um booleano indicando se a mensagem foi ou não enviada. Quando uma mensagem não é enviada

²¹ A seção 2.2 do segundo capítulo traz o preço médio global, aqui no Brasil apesar da realidade ser bem diferente as operadoras de telefonia tem criado pacotes que reduzem significativamente o custo do envio de mensagens SMS.

de imediato, ela é armazenada no banco de dados do paciente e é enviada junto com ele no momento de reportar o histórico do paciente.

Foi ainda configurado via *software* que o sistema execute automaticamente em intervalos de tempo pré-definidos. Esse intervalo entre uma inicialização e outra é ajustado de acordo com as ações (normal, alerta e emergência) ativadas pelo motor de inferência. Quanto mais grave a situação de alerta sobre o estado de saúde do usuário, menor o intervalo de tempo para a próxima execução.

A avaliação do aplicativo *Monitor Intelligent Prevention* foi realizada através de experimentos simulados em computador, com o uso de dados sintéticos imitando medições em um usuário e um conjunto de regras de produção construído por COPETTI (2010). Com estes experimentos foi possível testar o funcionamento do *software* que se mostrou bastante promissor no auxílio à assistência médica remota.

O aplicativo apresentado neste trabalho é um MIDlet de acompanhamento de saúde de pacientes cardiovasculares que, com recursos de inteligência artificial personalizados para o seu usuário, torna este processo mais eficaz e confortável ao mesmo, bem como auxilia os profissionais de saúde no diagnóstico de alguma situação de saúde e no processo de tomada de decisão quanto a algum tratamento, por armazenar as informações do histórico de saúde do usuário do sistema.

7.1 Discussões

Utilizando técnicas de teleassistência, inteligência artificial e desenvolvimento de software embarcado (ou desenvolvimento para *mobile*) é possível conseguir formas de custo-benefício para oferecer às pessoas com doenças crônicas uma centralização do serviço no usuário, reduzir a incerteza na hora do diagnóstico de determinados sintomas e melhorar o contato com os serviços assistência à saúde. Unindo aspectos destas três áreas, o *MonitorPrevIntel* é um sistema para suporte à prevenção de doenças cardiovasculares utilizando monitoramento remoto inteligente. Para fazer um paralelo entre as soluções de sistemas de monitoramento remoto preventivo apresentadas no Capítulo 3 com o *MonitorPrevIntel*, a Tabela 7.1 elenca as principais características desses sistemas, expostas de maneira a simplificar a comparação entre os mesmos.

Tabela 7.1 - Síntese das características das principais soluções de sistemas de monitoramento remoto preventivo.

	<i>Body Sensors Network</i> (JAFARI <i>et al.</i> , 2007)	<i>TeleCardio</i> (ANDREÃO <i>et al.</i> , 2006)	<i>AMON</i> (ANLIKER <i>et al.</i> , 2004)	<i>Machado et al.</i> (2008)	Sistema proposto por Lee <i>et al.</i> (2006)	DMS (HERBERT <i>et al.</i> , 2006)	H-SAUDE (COPETTI, 2010)	EMERGE (XIAO; CHEN, 2008)	<i>MonitorPrevIntel</i>
Objetivo	Classificação de movimentos físicos	Monitoramento remoto de pacientes cardíacos	Monitoramento remoto de pacientes cardio-respiratórios	Monitoramento remoto de pacientes	Monitoramento remoto de pacientes hipertensos	Gerenciamento de dados flexível e efetivo em redes de sensores sem fio de pacientes	Monitoramento remoto ubíquo e inteligente	Monitoramento de idosos	Agente assistente personalizado para monitoramento remoto preventivo de doenças cardiovasculares.
Variáveis Consideradas	Temperatura, pressão arterial e condutância do nível da pele	Eletrocardiograma (ECG)	Taxa de pulso, oxigenação, pressão arterial, ECG, temperatura e acelerômetros	Ritmo cardíaco, temperatura corporal e pressão sanguínea	Pressão sanguínea, taxa de pulso, ECG	ECG, pressão sanguínea e taxa de pulso	Pressão e frequência cardíaca, atividade do paciente e temperatura ambiente	Movimento, posição horizontal, temperatura corporal e fotopletiografia	Frequência cardíaca e respiratória, pressão arterial, oxigenação e temperatura
Tipo de Sistema (Móvel ou Home care)	Móvel (Unidades de Processamento (UP) Local e Pocket PC, ou Celular)	Híbrido (Holter e Computador)	Móvel (Dispositivo próprio)	Móvel (Celular)	Móvel (Celular)	Híbrido (Computadores e PDAs)	Home care	Home care	Móvel (Celular)
Execução	Não explicitada	Alertas ativados pelo usuário	Contínuo						Contínua e automática
Mecanismo de Inteligência Remoto	Reconhecimento de padrões nas UP	Inexistente	Classificação com base em pré-definição	Inexistente	Classificação com base em pré-definição	Uso de ontologias e múltiplos agentes	Ontologias e lógica Fuzzy	Reconhecimento de padrões	Inferência com base em regras de produção
Comunicação com Central Médica	Internet	Através de um computador	SMS/GPRS e Internet	GPRS (GSM e TDMA)	SMS/GPRS	Não explicitado	Não explicitado	Não explicitado	SMS/GPRS
Linguagem de Programação	Não explicitada	Não explicitada	Java	Java	Java	Java	Não explicitada	Não explicitada	Java (JME)

Ao observar a tabela podemos identificar vários pontos em comum. Por exemplo, é interessante notar que a maioria dos sistemas implementados para monitoramento remoto utiliza a linguagem de programação Java. Bem como é notável o interesse na construção de projetos que auxiliem no acompanhamento cardíaco do paciente independente de sua localização, com os sistemas móveis.

Porém, ao comparar o *MonitorPrevIntel* às demais soluções para monitoramento remoto preventivo é possível destacar dois grandes avanços obtidos através deste trabalho. O primeiro, diz respeito ao uso de um mecanismo de inferência, baseado em um sistema de regras de produção, totalmente embutido em um telefone celular e personalizado para cada usuário. E o segundo, a ubiquidade de um sistema de monitoramento de saúde, ao usar o telefone celular como um instrumento autônomo inteligente no acompanhamento e análise de dados de saúde.

Ao analisar as características Tipo de Sistema, Mecanismo de Inteligência Remota e Comunicação com Central Médica, destaca-se a grande diferença entre as arquiteturas propostas pelas diversas soluções de monitoramento existentes. Além desse fato, essas três características também auxiliam na afirmação dos avanços do *MonitorPrevIntel* apontados anteriormente. Com base nessas características é possível comprovar os fatos apresentados no Capítulo 1 na seção 1.3 (Justificativas), os quais afirmam que, de maneira geral, ou os sistemas de monitoramento remoto criam ambientes domiciliares como extensão dos ambientes hospitalares, ou ao optarem por utilizar um dispositivo pervasivo, tal como o telefone celular, não utilizam seu poder de processamento, subutilizando-o para roteamento de dados.

Comparando a solução apresentada neste trabalho com o *Body Sensor Network* (JAFARI *et al.*, 2007) observa-se a utilização do dispositivo móvel, *Pocket PC* ou telefone celular, apenas para armazenamento dos dados e envio dos mesmos através de uma conexão de Internet. No *Body Sensor Network* ainda é utilizado algum mecanismo de inteligência, mas esse é realizado no dispositivo responsável por aferir os dados de entrada e não no telefone móvel. No sistema proposto por MACHADO *et al.* (2008), novamente o celular é utilizado apenas com o objetivo de rotear as informações aferidas por sensores para uma central médica.

Contrapondo-se a esses dois sistemas, o *MonitorPrevIntel* tem como foco o uso do aparelho celular como um instrumento de monitoramento móvel inteligente, que processe, analise e tome decisões sobre a detecção de situações anormais nas condições de saúde de seu

usuário. Nesse *software*, o envio de informações para uma central médica é só mais uma de suas funções.

O sistema apresentado como TeleCardio (ANDREÃO *et al.*, 2006) também se preocupa com o acompanhamento dos sinais cardíacos de seu usuário, porém a percepção de uma situação crítica não se dá em tempo real. Os dados aferidos nesse sistema só são enviados ao centro médico quando o usuário conecta o sistema a um computador para transferir os dados através da internet e é o usuário que se responsabiliza por ativar alertas quando detectada alguma situação de anormalidade de saúde.

No *MonitorPrevIntel* e no AMON (ANLIKER *et al.*, 2004) logo após a leitura dos sinais vitais é verificada a condição de saúde do paciente e, caso o mecanismo de inteligência do sistema ache necessário, é ativado algum alerta. Ambos os sistemas agem de maneira contínua a ler e verificar a necessidade de emissão de alertas ou outras ações. A diferença entre eles é que o AMON é um sistema composto por um dispositivo móvel próprio. É nele que reside o mecanismo de inteligência. Já o *MonitorPrevIntel* utiliza um dispositivo do cotidiano para implantar um mecanismo de inteligência trazendo ubiquidade ao sistema. Além disso, esse último sistema também realiza o monitoramento continuamente e com ativação controlada pelo mecanismo de inteligência, tornando o aplicativo independente da intervenção do usuário para sua execução.

O sistema proposto por LEE *et al.* (2006) e o DMS (HERBERT *et al.*, 2006) também se caracterizam por realizar algum tipo de processamento no dispositivo remoto. Porém o primeiro, apesar de utilizar o celular para algum processamento de dados realiza apenas algumas classificações com base em valores pré-definidos. O segundo sistema já utiliza mecanismos mais rebuscados de inteligência, porém utiliza dispositivos com maior poder computacional que os telefones móveis, tais como os PDA e os computadores. Já o *MonitorPrevIntel* utiliza mecanismos de inteligência totalmente embutidos no telefone celular a partir de um sistema de inferência que toma por base regras de produção.

Em relação ao objetivo proposto por cada solução, apesar da aparente divergência, é possível observar que elas se dividem em três linhas principais: as voltadas ao monitoramento com base em sinais cardíacos, as que se dedicam a observar a movimentação como elemento determinante na detecção de uma situação crítica para o paciente e, por último, as soluções que pretendem apresentar uma estrutura de propósito geral. Apesar dessas diferenças nota-se, através das variáveis consideradas, que todas as soluções utilizam ao menos uma medida cardíaca, ressaltando a importância da análise dos sinais vitais relacionados ao coração para a análise do estado de saúde do usuário do sistema. Em adição, eles têm como preocupação dar

feedback para algum centro de supervisão médica a respeito da condição de saúde do seu usuário para que especialistas em saúde acompanhem-no.

7.2 *Trabalhos Futuros*

Os avanços apresentados por este trabalho demonstram que ainda há muita tecnologia disponível a ser explorada para fins de telemedicina e de telemonitoramento e que o *MonitorPrevIntel* pode ser expandido em diversas direções. Primeiramente, é importante destacar que o sistema apresentado deve ser testado em diferentes aparelhos de telefonia móvel, para verificar sua real portabilidade. Isto não pode ser realizado durante a vigência do trabalho devido à ausência de financiamento para compra de equipamentos, que também inviabilizou os testes utilizando sensores reais de sinais vitais. O sistema também deve ter seu uso avaliado em ambientes reais no monitoramento de muitos pacientes reais. Isto permitirá (1) a verificação, de maneira mais ampla, da aplicabilidade do aplicativo, (2) a construção de um banco de dados contendo monitoramento diário de sinais vitais de diversos pacientes e, (3) a construção, em parceria com especialistas médicos, de novos conjuntos de regras de produção sobre acompanhamento de pacientes com doenças cardiovasculares.

Outra modificação interessante seria implementar a execução ou outras ações do sistema através do recebimento de uma ação externa, como o recebimento de uma mensagem SMS, e o uso de um módulo de localização geográfica. Com isso o paciente poderia ser localizado quando detectada uma situação de emergência e o médico poderia pedir uma inicialização extraordinária do aplicativo ou ainda enviar procedimentos para serem seguidos pelo paciente, de forma que ele controlasse uma situação de alerta. O sistema poderia também ser modelado para ter como entrada um plano de cuidados do paciente e a partir deste notificar o paciente sobre horários de medicamentos e recomendações de dietas, o que necessitaria uma conexão através de uma operadora de telefonia móvel.

A estrutura criada para o aplicativo não é totalmente flexível para incorporação de novas regras de produção por limitações da linguagem JME que não permite a reflexão (acesso a informações da classe – atributos, métodos e construtores – em tempo de execução). Outra proposta futura é a construção deste sistema para os aparelhos celulares com sistema operacional Android. Isto porque a linguagem Java para Android tem um conjunto bem menos restrito que o JME (por exemplo: o uso de reflexão). Deste modo, teríamos um maior

controle sobre os dispositivos e ainda a execução de aplicativos em *background*. Estas características mais amplas de possibilidades do Java para Android permitiriam que o aplicativo fosse executado em tempo real e que se tornasse flexível o suficiente para seu código se modelar conforme o conjunto de regras de produção utilizado. Tal fato permitiria que o aplicativo pudesse até ser adequado para incluir novos sinais vitais e atender a outros grupos que necessitam de acompanhamento de saúde, como os idosos.

Por fim, seria bastante interessante transformar esse sistema em uma plataforma multi-agente para monitoramento remoto preventivo de doenças cardiovasculares o que poderia ser alcançado integrando o aplicativo desenvolvido neste trabalho com três novos sistemas: um sistema que auxiliasse os médicos na geração da base de regras de produção, um sistema que realizasse mecanismos de inteligência mais robustos sobre os dados dos pacientes e um *driver* Bluetooth para comunicação de sensores, que utilizam o perfil Bluetooth *Health Device Profile* (HDP), com o telefone celular.

O sistema de geração de base de regras de produção seria uma interface na qual o médico pudesse criar o arquivo personalizado de regras de produção para os pacientes de maneira abstraída da linguagem de escrita da base de conhecimento. Por exemplo, escolhendo opções em listas de itens e completando parâmetros numéricos. O segundo sistema seria responsável por receber os dados reportados pelo *MonitorPrevIntel*, armazená-los em uma base central de informações e, sobre aqueles, realizar mecanismos de inteligência mais robustos para extração de novas informações, tal como sugerir aos especialistas médicos a adição de novas regras de produção com base no histórico de acompanhamento diário dos sinais vitais dos pacientes. O último, o driver Bluetooth HDP para telefones móveis, serviria de interface de comunicação Bluetooth entre os sensores de medição dos sinais vitais e o aplicativo móvel *MonitorPrevIntel*.

Referências

- AHMEDA, S. S.; EDWILA, A. M. A. **Secure Protocol for Short Message Service**. World Academy of Science, Engineering and Technology. [S. l.] , 2009. v. 49, p. 864.
- ALBUQUERQUE, R. *et al.* **Embedding J2ME-based inference engine in handheld devices: The KEOPS case study**. In: Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices, 1., 2002, Bologna, IT. Proceedings of the Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices, Bologna: [s. n.], 2002. p. 4.
- ALBUQUERQUE, R. **kAgent: Uma API Java para Agentes Inteligentes em Dispositivos Móveis**. 2002. 140f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, PE. 2002.
- ALECRIM, E. **Tecnologia Bluetooth**. InfoWester, São Paulo, 16 mai. 2011. Disponível em: <<http://www.infowester.com/bluetooth.php>>. Acesso em: 17 mai. 2011.
- ANDREÃO, R. V.; PEREIRA FILHO, J. G.; CALVI, C. **TeleCardio: Telecardiologia a serviço de pacientes hospitalizados em domicílio**. In: Congresso Brasileiro de Informática em Saúde (CBIS), 10., 2006, Florianópolis, SC. Anais... Florianópolis, SC: Sociedade Brasileira de Informática em Saúde (SBIS), 2006.
- ANLIKER, U. *et al.* **AMON: A Wearable Multiparameter Medical Monitoring and Alert System**. IEEE Transactions on Information Technology in Biomedicine, Zurich, Suíça, 2004. v. 8, n. 4, p. 415-427.
- AYABE, N. B. S.; CHANDER, W. S. S.; MIZIKOVSKY, M. S. B. **Short Message Service**. US006141550A, 8 jan. 1999, 31 out. 2000.
- BAADER, F. **Logic-Based Knowledge Representation**. In: WOOLDRIDGE, M. J.; VELOSO, M. Lecture Notes in Computer Science: Artificial Intelligence Today, Recent Trends and Developments, 1. ed. Alemanha: Springer Verlag, 1999. v. 1600/1999, p. 13-41. Disponível em: <<http://lat.inf.tu-dresden.de/research/papers/1999/Baader-LNAI-1999.ps.gz>>. Acesso em: 10 mai. 2011.
- BARBOSA, T. M. G. A. *et. al.* **Sistema Pessoal Móvel de Monitoração da Saúde: Algoritmo para captura inteligente de sintomas**. In: Congresso Brasileiro de Informática em Saúde (CBIS), 9., 2004, Ribeirão Preto, SP. Anais... Ribeirão Preto, SP: Sociedade Brasileira de Informática em Saúde (SBIS), 2004. p. 321-326.
- BLUETOOTH SIG. **A Look at the Basics of Bluetooth Wireless Technology**. The official Bluetooth® technology website, [S. l.], [2011]. Disponível em: <<http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>>. Acesso em: 31 mai. 2011

BLUETOOTH SIG. **Bluetooth Core Specification**. Version 4.0. [S.l.], [2010]. Disponível em: <https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737>. Acesso em: 08 jun. 2011.

BLUETOOTH SIG. **Building with the Technology: Overview**. The official Bluetooth SIG member website, [S.l.], [2011]. Disponível em: <<https://www.bluetooth.org/Building/overview.htm>>. Acesso em: 02 jun. 2011.

BOEHM, B. **A Spiral Model of Software Development and Enhancement** - *IEEE Computer*, vol.21, 5, May 1988, pp 61-72

BOUNCY Castle. **The Legion of the Bouncy Castle**. [S. l.]: bouncycastle.org, [20??]. Disponível em: <<http://www.bouncycastle.org/>>. Acesso em: 12 dez. 2012.

BRASIL fecha 2011 com recorde de habilitações na telefonia móvel. ANATEL, Brasília, 16 jan. 2012. Disponível em: <<http://www.anatel.gov.br/Portal/exibirPortalPaginaEspecialPesquisa.do?acao=&tipoConteudoHtml=1&codNoticia=24506>>. Acesso em: 29 fev. 2011.

CÂMARA, M. S. A. L. **Inteligência Artificial: Representação de Conhecimento**. [2000 ou 2001]. 8 f. Trabalho apresentado como requisito parcial para aprovação na disciplina Disciplina de Comunicação Técnica Profissional, Universidade de Coimbra, Coimbra, PT, [2000 ou 2001]. Disponível em: <student.dei.uc.pt/~mcamara/artigos/inteligencia_artificial.pdf>. Acesso em: 05 abr. 2011.

CARDOSO, J. **Java para Telemóveis MIDP 2.0**. Porto, Portugal: FEUP Edições, 2007. Disponível em: <http://books.google.com.br/books?id=EhPnsGCyPXUC&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em: 29 fev. 2012.

CARVALHO, M. A. S. **Um sistema de monitoramento remoto de pacientes usando rede sem fio**. 2005. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2005.

CARVALHO, S. T. *et al.* **Monitoramento Remoto de Pacientes em Ambiente Domiciliar**. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 28., 2010, Gramado, RS. Anais... Gramado: Sociedade Brasileira de Computação (SBC), 2010. P. 1005-1012.

CASTRO, A. M. *et al* O. **Redes de Sensores Sem Fio (RSSF)**. 2010. Trabalho apresentado como requisito parcial para aprovação na disciplina Redes de Computadores I, Grupo de Teleinformática e Automação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010. Disponível em: <http://www.gta.ufrj.br/grad/10_1/rssf/index.html>. Acesso em: 16 mai. 2011.

CHANGE Vision. **Astah Community**. [S. l.]: Change Vision, 2012. Disponível em: <<http://astah.net/editions/community>>. Acesso em: 3 fev. 2012.

COBERTURA das Operadoras e População Atendida. Teleco: Inteligência em Comunicações, São José dos Campos, SP, 07 jun. 2011. Disponível em: <<http://www.teleco.com.br/cobertura.asp>>. Acesso em: 28 jun. 2011.

COPETTI, A. **Monitoramento Inteligente e Sensível ao Contexto na Assistência Domiciliar Telemonitorada.** 2010. 135 f. Tese (Doutorado em Computação) – Universidade Federal Fluminense, Niterói, RJ, 2010.

COSTA, M. T. C. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância.** 1999. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, SC, 1999.

CRUZ, E. S.; PEREIRA, G. C. C. F.; SILVA, R. R. **Sistema de SMS seguro.** 2008. Trabalho de conclusão de curso (Graduação em Engenharia de Computação) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.

ESTIMATED total deaths ('000), by cause and WHO Member State, 2004 (a). World Health Organization, [S. l.], [2004]. Disponível em: <http://www.who.int/entity/healthinfo/global_burden_disease/gbddeathdalycountryestimates2004.xls>. Acesso em: 27 jun. 2011

ETSI. **Alphabets and language-specific information (GSM 03.38).** 17 f. Valbonne, FR: ETSI TC-SMG, 1996. Disponível em: <http://www.mobilecity.cz/doc/GSM_03.38_5.3.0.pdf>. Acesso em: 15 jun. 2011.

ETSI. **Technical realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 03.40).** 109 f. Valbonne, FR: ETSI TC-SMG, 1996. Disponível em: <http://www.mobilecity.cz/doc/GSM_03.40_5.3.0.pdf>. Acesso em: 15 jun. 2011.

FACTSHEET N° 137: Cardiovascular diseases (CVDs). World Health Organization, [S. l.], jan. 2011. Disponível em: <<http://www.who.int/mediacentre/factsheets/fs317/en/index.html>>. Acesso em: 27 jun. 2011.

GASCÓN, D. **802.15.4 vs ZigBee.** Wireless Sensor Networks, [S.l.], 17 nov. 2008. Disponível em: <<http://sensor-networks.org/index.php?page=0823123150>>. Acesso em: 20 mai. 2011.

GATTO, E. C. **Sistema de Apoio Policial.** 2007. Monografia (Engenharia da Computação) – Unifev/Centro Universitário de Votuporanga, Votuporanga, SP, 2007. Disponível em: <<http://javanoroeste.com.br/javanoroeste/novo/artigos/javame.html>>. Acesso em: 10 mar. 2012.

HAYUN, R. B. **Java ME on Symbian OS.** England: John Wiley & Sons Ltd, 2009.

HERBERT, J. *et al.* **Mobile agent architecture integration for a wireless sensor medical application.** In: IEEE International Workshop on Intelligent Agents in Wireless Sensor Networks (IA-WSN), 1., 2006, Hong Kong. Proceedings of First IEEE International

Workshop on Intelligent Agents in Wireless Sensor Networks (IA-WSN). Hong Kong: IEEE, 2006.

HPY BTHDY TXT! BBC News, UK, 03 dez. 2002. Disponível em: <http://news.bbc.co.uk/2/hi/uk_news/2538083.stm>. Acesso em: 09 jun. 2011.

JAFARI, R. *et al.* **Platform Design for Health-care Monitoring Applications**. In: Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS, 2., 2007, Boston, USA. Proceedings of Workshop on High Confidence Medical Devices, Software, and Systems (HCMDSS)'07. Boston, USA: Medical Device Plug-and-Play (MD PnP) Interoperability Program, 2007. p. 88-94.

JOVANOVIĆ, E. *et al.* **Patient Monitoring Using Personal Area Networks of Wireless Intelligent Sensors**. Biomedical Sciences Instrumentation, [S. l.], 2001. v. 37, p. 373-378.

LEE, R. G. *et al.* **A mobile-care system integrated with Bluetooth blood pressure and pulse monitor, and cellular phone**. IEICE Transactions Information and Systems, Tokio, JP, v. E89-D, n. 5, p. 1702-1711, mai. 2006.

LEITE, J. C. **O Modelo Espiral**. 2007. Disponível em: <<http://engenhariadesoftware.blogspot.com/2007/03/o-modelo-espiral.html>>. Acesso em: 06 mai. 2012.

LIEBERMAN, H.; MASON, C. **Intelligent agent software for medicine**. Stud Health Technol Inform'02, USA, n. 80, p. 99-109, 2002.

LO, K. F.; LUO, J.; YEOH, S. K. **Secure short message service**. WO 2009/154580 A1, 19 jun. 2009, 23 dez. 2009.

MABRY, S. L. *et al.* **Intelligent agents for patient monitoring and diagnostics**. In: Association for Computing Machinery (ACM) International Symposium on Applied Computing, 18., 2003, Melbourne, USA. Proceedings of the 2003 ACM symposium on Applied computing. Melbourne, USA: Association for Computing Machinery (ACM), 2003.

MACHADO, A. *et al.* **Utilização de Dispositivos Móveis, Web Services e Software Livre no Monitoramento Remoto de Pacientes**. In: Congresso Brasileiro de Informática em Saúde (CBIS), 11., 2008, Campos do Jordão, SP. Anais... Campos do Jordão, SP: Sociedade Brasileira de Informática em Saúde (SBIS), 2008.

MARÇAL, I. S. **Bluetooth e Zigbee um padrão para redes pessoais sem fio**. 2008. 49 f. Trabalho de Conclusão de Curso (Especialização) – Departamento de Computação, Universidade Estadual de Londrina, Londrina, PR, 2008.

MARDIKAR, U. **Secure short message service (SMS) communications**. WO 2009/149376 A1, 05 jun. 2009, 10 dez. 2009.

MAREJKA, R. **Learning Path: MIDlet Life Cycle**. 2005. Oracle Corporation - Sun Developer Network, [S. l.], fev. 2005. Disponível em: <<http://developers.sun.com/mobility/learn/midp/lifecycle/>>. Acesso em: 20 abri. 2012.

MARKS, R. A et al. **Multimídias Móveis**. Campo Grande, MS: Universidade Federal de Mato Grosso do Sul, 2009. Disponível em: <<http://www.slideshare.net/phpmobile/apresentacao-multimidia-mveis?type=presentation>>. Acesso em: 17 jan. 2010.

MEA, V. D. **Agents acting and moving in healthcare scenario: A paradigm for telemedical collaboration**. IEEE Transactions on Information Technology in Biomedicine, Zurich, Suíça, 2001. v. 5, n. 1, p 10-13.

MELLO JÚNIOR, J. P. **Bluetooth 4.0 recebe aprovação e já pode ser utilizado pela indústria**. IDG News Service, [S. l.], jul. 2010. Disponível em: <<http://idgnow.uol.com.br/internet/2010/07/07/bluetooth-4-0-recebe-aprovacao-e-ja-pode-ser-utilizado-pela-industria/>>. Acesso em: 06 jun. 2011.

MENEZES JÚNIOR, J. V. et al. **InteliMed: uma experiência de desenvolvimento de sistema móvel de suporte ao diagnóstico médico**. Revista Brasileira de Computação Aplicada, Brasil, 2011. v. 3, n. 1.

MOKHTAIAN, K. **A study of the short message service of a nationwide cellular network: a summary**. 2007. 4 f. Trabalho apresentado como requisito parcial para aprovação na disciplina Selected Topics on Multimedia Networking, School of Computing Science, Simon Fraser University, Burnaby, CA, 2007. Disponível em: <<http://www.cs.sfu.ca/~jcliu/cmpt880-2007/7.pdf>>. Acesso em: 26 jun. 2011.

MONSIGNORE, F. **Sensoriamento de ambiente utilizando o padrão ZigBee**. 2007. 74 f. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2007.

MURCHOW, J. W. **Core J2ME: Tecnologia e MIDP**. São Paulo: Pearson-Makron Books, 2004.

NAVEGA, S. **Inteligência Artificial, Educação de Crianças e o Cérebro Humano**. Revista de Estudos de Comunicações of the University of Santos, Santos, 2000. Ano 25, n. 72, p. 87-102. Disponível em: <<http://www.intelliwise.com/reports/p4port.htm>>. Acesso em: 5 abri. 2012.

NEWELL, A. **Physical Symbol Systems**. Cognitive Science Journal, [S.l.], 1980. Volume 4, Issue 2, p. 135 - 183, Abril 1980. Disponível em: <http://hci.ucsd.edu/102a/readings/NewellPSSH.pdf>>. Acesso em: 20 dez. 2011.

NÚMERO de celulares no Brasil cresce 9,6 milhões em 2011. Portal Brasil, Brasília, 25 mai. 2011. Disponível em: <<http://www.brasil.gov.br/noticias/arquivos/2011/05/25/numero-de-celulares-no-brasil-cresce-9-6-milhoes-em-2011>>. Acesso em: 28 jun. 2011.

OMG. **UML® Resource Page**. [S. l.]: Object Management Group, Inc, [1997-2012]. Disponível em: <<http://www.uml.org/>>. Acesso em: 3 mar. 2012.

ORACLE. **Java Card Technology Overview**. [S. l.]: Oracle Corporation, [20??]. Disponível em: <<http://www.oracle.com/technetwork/java/javacard/overview/overview-jsp-135353.html>>

PALAZZO, L. A. M. **Inteligência Artificial: Sistemas multiagentes**. Pelotas, RS: Universidade católica de Pelotas, 2005. Disponível em: <www.ia.ucpel.tche.br/~lpalazzo/Aulas/IA/m03/m03.ppt>. Acesso em: 12 mai. 2011.

PATTICHIS, C. S. et al. **Wireless telemedicine systems: an overview**. IEEE Antennas and Propagation Magazine, [S. l.], 2002. v. 44, n. 2, p. 143-153.

PETTERSSON, L. **SMS and the PDU format**. Dream Fabric, [S. l.], [2005] Disponível em: <<http://www.dreamfabric.com/sms/>>. Acesso em: 12 abr. 2011.

PINHEIRO, J. M. S. **ZigBee em Home Area Network**. Projeto de Redes, [S.l], 12 fev. 2006. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_zigbee_em_home_area_network.php>. Acesso em: 22 mai. 2011

PIZARRO, P. J. C. et al. **Monitoramento remoto de sinais bioelétricos**. In: Congresso Latinoamericano de Ingeniería Biomédica, 2., 2001, La Habana, Cuba. Memorias... La Habana, Cuba: Sociedad Cubana de Bioingeniería, 2001.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: A modern approach**. 2. New Jersey, USA: Prentice Hall (Pearson Education), 2003.

SANTOS, S. T. **Redes de Sensores sem Fio em Monitoramento e Controle**. 2007. 71 f. Dissertação (Mestrado em Ciências em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

SHAH, S. et al. **Zip it up SMS “A path breaking model in the field of mobile messaging”**. In: IEEE International Conference on Computer Science and Information Technology (IEEE ICCSIT), 3., 2010, Chengdu, CHN. Proceedings of the 2010 3rd IEEE International Conference on Computer Science and Information Technology. Chengdu, CHN: IEEE, 2010. p. 119-125.

SIQUEIRA, E. **Tecnologias que mudam nossa vida**. São Paulo: Saraiva, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Addison-Wesley. 2007.

STEIN, P. **BlueTooth**. 2003. Trabalho apresentado como requisito parcial para aprovação na disciplina Redes de Computadores, Grupo de Teleinformática e Automação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003. Disponível em: <http://www.gta.ufrj.br/seminarios/semin2003_1/stein/>. Acesso em: 18 mai. 2011.

TEIXEIRA, I. M. **Considerações de QoS para telemedicina e e-health em redes sem fios.** 2009. Trabalho apresentado como requisito parcial para aprovação na disciplina Tópicos Especiais: Mecanismos e Arquiteturas de Teleinformática, Pós-graduação em Modelagem Computacional, Laboratório Nacional de Computação Científica (LNCC) do Ministério da Ciência e Tecnologia, Petrópolis, RJ, 2009. Disponível em: <<http://wiki.martin.lncc.br/atagomes-cursos-lncc-gb500-20094/file/gb500-iuri.pdf>>. Acesso em: 28 jun. 2011.

TRAJANO, E.; ROBIN, J. **Engenharia do Conhecimento.** 2003. Trabalho apresentado como requisito parcial para aprovação na disciplina Métodos de Computação Inteligentes, Mestrado Acadêmico do Centro de informática, Universidade Federal de Pernambuco, Recife, PE, 2003. Disponível em: < <http://www.cin.ufpe.br/~in1006/2003/KEAM.ppt>>. Acesso em: 16 mai. 2011.

VOSKARIDES, S. et al. **Mobile health systems: a brief overview.** In: SPIE Digital Wireless Communications, 4., 2002, Orlando, USA. Proceedings... Orlando, USA: SPIE, 2002. v. 4740, p. 124-131.

WOCHNER, C. R. **Agente médico para tecnologia móvel.** 2005. 51 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo, RS, 2005.

WOODWARD, B.; ISTEPANIAN, R.; RICHARDS, C. **Design of a telemedicine system using a mobile telephone.** IEEE Transactions on Information Technology Biomedicine, United Kingdom, 2001. v. 5, n. 1, p. 13–15.

XIAO, Y; CHEN, H. **Mobile Telemedicine: A computing and networking perspective.** Boca Raton, USA: Taylor & Francis Group (LLC), 2008. cap. 1. p. 5-25.

ZERFOS, P et al. **A study of the short message service of a nationwide cellular network.** In: Internet Measurement Conference, 6., 2006, Rio de Janeiro. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. Rio de Janeiro: ACM, 2006. p. 263-268

ZIGBEE Alliance. **About Technology** ZigBee Technology, [S. l.], [2011]. Disponível em: <<http://www.zigbee.org/About/AboutTechnology/ZigBeeTechnology.aspx>>. Acesso em: 22 mai. 2011.

ZIGBEE Standards Organization. **ZigBee Specification.** [S. l.]: ZigBee Alliance, 2008. Disponível em: < <http://www.zigbee.org/>>. Acesso em: 22 mai. 2011.

Apêndice A – Classificação das variações dos Sinais Vitais

A classificação das variações dos sinais vitais dos pacientes como normais ou como situação de saúde crítica pode ser feita de diferentes maneiras, a fórmula utilizada nesse trabalho leva em consideração o valor normal do sinal vital do usuário e uma tabela de intervalos de referência, retirado de COPETTI (2010). O valor normal do sinal vital seja frequência cardíaca, frequência respiratória, pressão arterial, oxigenação do sangue, temperatura corporal ou qualquer outra é o valor médio que o paciente deve apresentar quando estiver em suas condições de saúde normal.

Para saber se o valor de um sinal vital de um determinado momento está dentro dos parâmetros de normalidades para o paciente esse dado é subtraído do valor normal do sinal e comparado com a tabela de referência. A diferença, o desvio, entre aquelas duas medidas deve ser enquadrada em uma das cinco categorias: MUITO_BAIXO, BAIXO, NORMAL, ALTO e MUITO_ALTO. Essa categorização é feita para que o trabalho use as regras de produção disponibilizadas por COPETTI (2010) e leva em conta a atividade física que está sendo realizada pelo usuário, mas caso o sistema utilize regras de comparação direta entre os valores o *MonitorPrevIntel* também oferece suporte a elas.

Essas cinco categorias são faixas de valores dentro das quais a diferença, entre o sinal em determinado momento e o sinal na condição normal da saúde do paciente, deve ser enquadradas. Os valores de cada categoria de classificação variam de acordo com a atividade física que o paciente estiver realizando. A Tese de COPETTI (2010) traz os dados para tabela de referência da variável pressão arterial sistólica, apresentados na Tabela A.1.

Tabela A.1 - Quadro de referência para a classificação do desvio da pressão arterial sistólica.

Atividade	Desvio Pressão Arterial Sistólica	Classificação
Dormindo	$]-\infty, -50]$	MUITO_BAIXO
Dormindo	$]-50, -20]$	BAIXO
Dormindo	$]-20, 0]$	NORMAL
Dormindo	$]0, 20]$	ALTO
Dormindo	$]20, +\infty [$	MUITO_ALTO
Atividade Doméstica	$]-\infty, -50]$	MUITO_BAIXO
Atividade Doméstica	$]-50, -5]$	BAIXO
Atividade Doméstica	$]-5, 10]$	NORMAL
Atividade Doméstica	$]10, 40]$	ALTO
Atividade Doméstica	$]40, +\infty [$	MUITO_ALTO
Repouso	$]-\infty, -50]$	MUITO_BAIXO
Repouso	$]-50, -10]$	BAIXO
Repouso	$]-10, 5]$	NORMAL
Repouso	$]5, 20]$	ALTO
Repouso	$]20, +\infty [$	MUITO_ALTO

A Tese de COPETTI (2010) traz ainda os valores de referência para classificação da frequência cardíaca, Tabela A.2, o valor a ser comparado com esses intervalos é o próprio sinal vital em determinado momento. Os dados de classificação apresentados nesse apêndice são os valores utilizados pelos casos de testes realizados para validação do sistema.

Tabela A.2 - Intervalos para classificação da frequência cardíaca.

Sinal Vital	Intervalo	Classificação
Frequência Cardíaca	]-∞, 60]	BAIXO
Frequência Cardíaca	]60, 100[	NORMAL
Frequência Cardíaca	[100, +∞[	ALTO

Apêndice B – Base de Conhecimentos do Caso de Teste

Base de conhecimento formada pelas regras de produção utilizada pelo caso de teste descrito no capítulo 6.

ruleBase teste

```
rule primeira:
  conditions:
    SBPVariancesSleeping = MUITO_BAIXO;
  actions:
    emergency;

rule segunda:
  conditions:
    SBPVariancesSleeping = MUITO_ALTO;
  actions:
    emergency;

rule terceira:
  conditions:
    SBPVariancesSleeping = ALTO;
  actions:
    alert;

rule quarta:
  conditions:
    SBPVariancesSleeping = BAIXO;
  actions:
    alert;

rule quinta:
  conditions:
    SBPVariancesSleeping = NORMAL;
  actions:
    normal;

rule sexta:
  conditions:
    heartRate = ALTO;
  actions:
    alert;

rule setima:
  conditions:
    heartRate = BAIXO;
  actions:
    alert;

rule oitava:
  conditions:
    heartRate = NORMAL;
  actions:
    normal;

rule nona:
  conditions:
    SBPVariancesStand = MUITO_ALTO;
    SBP = hypertensive;
  actions:
    emergency;
```

.

Apêndice C – Testes

Para apresentar uma amostra da execução dos testes, foi elaborada a Tabela C.2 que traz vinte e sete casos de teste com os dados de entrada, um conjunto de medições – cujos dados representam Frequência Cardíaca (FC), Frequência Respiratória (FR), Pressão Arterial Sistólica (PAS), Pressão Arterial Diastólica (PAD), Oxigenação no Sangue (OS), Temperatura Corporal (TC) –, um conjunto de regras (Base de Regras) que comporá a base de conhecimento do MonitorPrevIntel e o estado atual do paciente (Sleeping, Stand, HomeActivity), e os dados de saída, que informam as ações executadas pelo agente. As regras utilizadas para construção das Bases de Regras para cada caso testado são referenciadas segundo o Número de Referência observado Tabela C.1.

Tabela C.1 - Regras utilizadas para teste do MonitorPrevIntel.

Regra	Nível de Prioridade (0: menor importância 1: maior importância)	Número para Referência
rule sleepMB: conditions: SBPVarianceSleeping = MUITO_BAIXO; actions: emergency;	1	1
rule sleepMA: conditions: SBPVarianceSleeping = MUITO_ALTO; actions: emergency;	1	2
rule standMA: conditions: SBPVarianceStand = MUITO_ALTO; SBP = hypertensive; actions: emergency;	1	3
rule standMB: conditions: SBPVarianceStand = MUITO_BAIXO; actions: emergency;	1	4
rule homeMA: conditions: SBPVarianceHomeActivity = MUITO_ALTO; SBP = hypertensive; actions: emergency;	1	5

continua

Tabela C.1 - Regras utilizadas para teste do MonitorPrevIntel.

continuação

Regra	Nível de Prioridade (0: menor importância 1: maior importância)	Número para Referência
rule homeMB: conditions: SBPVarianceHomeActivity = MUITO_BAIXO; actions: emergency;	1	6
rule sleepA: conditions: SBPVarianceSleeping = ALTO; actions: alert;	0,5	7
rule standA: conditions: SBPVarianceStand = ALTO; actions: alert;	0,5	8
rule homeA: conditions: SBPVarianceHomeActivity = ALTO; actions: alert;	0,5	9
rule sleepB: conditions: SBPVarianceSleeping = BAIXO; actions: alert;	0,3	10
rule standB: conditions: SBPVarianceStand = BAIXO; actions: alert;	0,3	11
rule homeB: conditions: SBPVarianceHomeActivity = BAIXO; actions: alert;	0,3	12
rule sleepN: conditions: SBPVarianceSleeping = NORMAL; actions: normal;	0,2	13
rule standN: conditions: SBPVarianceStand = NORMAL; actions: normal;	0,2	14

continua

Tabela C.1 - Regras utilizadas para teste do MonitorPrevIntel.

continuação

Regra	Nível de Prioridade (0: menor importância 1: maior importância)	Número para Referência
rule homeN: conditions: SBPVarianceHomeActivity = NORMAL; actions: normal;	0,2	15
rule heartA: conditions: heartRate = ALTO; actions: alert;	0,1	16
rule heartB: conditions: heartRate = BAIXO; actions: alert;	0,1	17
rule heartN: conditions: heartRate = NORMAL; actions: normal;	0,1	18
rule standMA1: conditions: SBPVarianceStand = MUITO_ALTO; SBP > 135; actions: emergency;	1	19
rule standMA2: conditions: SBPVarianceStand = MUITO_ALTO; DBP > 85; actions: emergency;	1	20
rule sleepFA: conditions: heartRate = ALTA; actions: alert;	0,5	21

Os casos de teste foram elaborados considerando os dados das variações de sinais vitais, constantes no Apêndice A, para que gerassem diferentes comportamentos para o agente do *MonitorPrevIntel*. Como o agente pode detectar diferentes situações de normalidade, alerta ou emergência o *MonitorPrevIntel* tem de ser configurado para três tempos distintos de execução automática. Desta maneira, para agilizar a execução dos testes, apresentados na Tabela C.2, foram escolhidos os tempos 30 segundos, 60 segundos e 120 segundos para as situações de emergência, alerta e normalidade, respectivamente.

Tabela C.1 - Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

Caso de Teste 1							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	1, 3, 5, 8, 9, 11, 12, 13, 14, 18, 19, 20, 21						
SAÍDAS:	Nenhuma regra ativada.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.	Ativação da regra 18: Normal detectado.	Nenhuma regra ativada.	Nenhuma regra ativada.	Ativação da regra 18: Normal detectado.
	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 2							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 oC						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	Stand	Stand	Stand	Stand	Stand	Stand	Stand
Base de regras	1, 3, 5, 8, 9, 11, 12, 13, 14, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 14: Normal detectado.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 14: Normal detectado.	Ativação da regra 14: Normal detectado.	Ativação da regra 14: Normal detectado.	Ativação da regra 14: Normal detectado.
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Ativação da regra 18: Normal detectado.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Ativação da regra 18: Normal detectado.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Medição e regras acionadas salvas no banco de dados local.
				Envio da base de dados local para o centro de saúde e limpeza da base local.			Execução agendada para 120 segundos.
				Execução agendada para 120 segundos.			

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 3							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 oC						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep
Base de regras	1, 3, 5, 8, 9, 11, 12, 13, 14, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra sleepdN: Normal detectado.	Nenhuma regra ativada.	Nenhuma regra ativada.	Ativação da regra 13: Normal detectado.	Ativação da regra sleepdN: Normal detectado.	Ativação da regra sleepdN: Normal detectado.	Ativação da regra 13: Normal detectado.
	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Ativação da regra 18: Normal detectado.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Ativação da regra 18: Normal detectado.
	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Medição e regras acionadas salvas no banco de dados local.
				Envio da base de dados local para o centro de saúde e limpeza da base local.			Execução agendada para 120 segundos.
				Execução agendada para 120 segundos.			Execução agendada para 120 segundos.

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 4							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 oC						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	2, 3, 4, 6, 7, 8, 9, 10, 11						
SAÍDAS:	Nenhuma regra ativada.	Ativação da regra homeA: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 5							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 oC						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	Stand	Stand	Stand	Stand	Stand	Stand	Stand
Base de regras	2, 3, 4, 6, 7, 8, 9, 10, 11						
SAÍDAS:	Nenhuma regra ativada.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 6							
ENTRADAS:							
Dados do usuário:	Usuario1, 45 anos, 90 kg, homem, 100 bpm, 16 ipm, 120/70 mmHg, 97% oxigenação, 37 oC						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	100	160	120	90	100	100	90
FR	16	18	17	15	16	16	15
PAS	120	140	130	120	120	120	120
PAD	70	90	80	80	80	70	70
OS	98	100	98	97	97	100	95
TC	36	36	36	36	36	37	37
Estado	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep
Base de regras	2, 3, 4, 6, 7, 8, 9, 10, 11						
SAÍDAS:	Nenhuma regra ativada.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 7							
ENTRADAS:							
Dados do usuário:	Usuario2, 35 anos, 62 kg, mulher, 90 bpm, 17 ipm, 120/80 mmHg, 99% oxigenação, 36 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	90	100	120	100	160	120	90
FR	15	16	17	16	18	17	15
PAS	120	120	130	120	120	130	140
PAD	70	70	80	70	80	80	80
OS	95	100	98	98	100	98	97
TC	37	37	36	36	36	36	36
Estado	HomeActivity	HomeActivity	Stand	Stand	Sleep	Sleep	Sleep
Base de regras	1, 3, 5, 7, 9, 10, 15, 21						
SAÍDAS:	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 8							
ENTRADAS:							
Dados do usuário:	Usuario3, 33 anos, 58 kg, mulher, 62 bpm, 14 ipm, 120/80 mmHg, 98% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	58	60	60	65	66	62	60
FR	14	15	14	15	15	15	14
PAS	110	120	125	130	127	125	120
PAD	75	78	80	85	82	80	80
OS	94	97	100	98	98	100	100
TC	36	37	37	37	37	37	37
Estado	Sleep	Stand	Stand	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	1, 3, 5, 7, 9, 10, 15, 21						
SAÍDAS:	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.
	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.
					Execução agendada para 120 segundos.		

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 9							
ENTRADAS:							
Dados do usuário:	Usuario4, 30 anos, 70 kg, homem, 80 bpm, 17 ipm, 120/80 mmHg, 91% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	80	80	82	84	85	85	80
FR	16	17	17	18	18	17	17
PAS	110	115	120	125	125	120	120
PAD	75	77	80	80	85	80	80
OS	90	90	90	93	91	91	92
TC	36	36	37	37	37	37	37
Estado	Stand	Stand	Stand	Stand	Stand	Stand	Stand
Base de regras	1, 3, 5, 7, 9, 10, 15, 21						
SAÍDAS:	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.
						Execução agendada para 120 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 10							
ENTRADAS:							
Dados do usuário:	Usuario2, 35 anos, 62 kg, mulher, 90 bpm, 17 ipm, 120/80 mmHg, 99% oxigenação, 36 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	90	100	120	100	160	120	90
FR	15	16	17	16	18	17	15
PAS	120	120	130	120	120	130	140
PAD	70	70	80	70	80	80	80
OS	95	100	98	98	100	98	97
TC	37	37	36	36	36	36	36
Estado	HomeActivity	HomeActivity	Stand	Stand	Sleep	Sleep	Sleep
Base de regras	1, 2, 4, 6, 16, 17						
SAÍDAS:	Nenhuma regra ativada.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Execução agendada para 120 segundos.
				Execução agendada para 60 segundos.			

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 11							
ENTRADAS:							
Dados do usuário:	Usuario3, 33 anos, 58 kg, mulher, 62 bpm, 14 ipm, 120/80 mmHg, 98% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	58	60	60	65	57	62	60
FR	14	15	14	15	15	14	14
PAS	110	120	125	130	127	125	120
PAD	75	78	80	85	82	80	80
OS	94	97	99	98	99	100	99
TC	36	37	37	37	37	37	37
Estado	Sleep	Stand	Stand	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	1, 2, 4, 6, 16, 17						
SAÍDAS:	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição e regra acionada salvas no banco de dados local.
	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.
				Execução agendada para 120 segundos.			

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 12							
ENTRADAS:							
Dados do usuário:	Usuario4, 30 anos, 70 kg, homem, 80 bpm, 17 ipm, 120/80 mmHg, 91% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	80	80	82	84	85	85	80
FR	16	17	17	18	18	17	17
PAS	110	115	120	125	125	120	120
PAD	75	77	80	80	85	80	80
OS	90	90	90	93	91	91	92
TC	36	36	37	37	37	37	37
Estado	Stand	Stand	Stand	Stand	Stand	Stand	Stand
Base de regras	1, 2, 4, 6, 16, 17						
SAÍDAS:	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.	Nenhuma regra ativada.
	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.
						Execução agendada para 120 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 13							
ENTRADAS:							
Dados do usuário:	Usuario5, 29 anos, 70 kg, homem, 68 bpm, 18 ipm, 110/80 mmHg, 99% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>		
FC	70	70	68	67	65		
FR	18	18	18	18	17		
PAS	125	125	120	115	115		
PAD	82	82	80	80	80		
OS	98	98	100	100	100		
TC	37	37	37	37	37		
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity		
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 15: Normal detectado	Nenhuma regra acionada	Nenhuma regra acionada		
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.		
	Execução agendada para 60 segundos	Execução agendada para 60 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos	Execução agendada para 120 segundos		
			Execução agendada para 120 segundos				

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 14							
ENTRADAS:							
Dados do usuário:	Usuario6, 50 anos, 70 kg, mulher, 95 bpm, 17 ipm, 120/85 mmHg, 97% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>			
FC	110	115	100	98			
FR	18	18	18	16			
PAS	135	140	130	120			
PAD	85	90	90	85			
OS	97	97	96	96			
TC	36	36	36	36			
Estado	Sleep	Stand	Stand	Stand			
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde	Nenhuma regra acionada			
	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde	Medição salva no banco de dados local.			
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Execução agendada para 120 segundos			
	Execução agendada para 60 segundos	Execução agendada para 60 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.				
			Execução agendada para 60 segundos				

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 15							
ENTRADAS:							
Dados do usuário:	Usuario7, 40 anos, 63 kg, mulher, 95 bpm, 17 ipm, 110/75 mmHg, 97% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	
FC	90	90	95	95	95	100	
FR	17	17	16	16	18	18	
PAS	110	110	115	120	120	120	
PAD	78	80	80	83	85	85	
OS	96	96	96	96	97	97	
TC	37	37	37	37	37	37	
Estado	Stand	Stand	Stand	Stand	Stand	Stand	
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 14: Normal detectado	Ativação da regra 14: Normal detectado	Ativação da regra 14: Normal detectado	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde	
	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde	
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	
	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos	Execução agendada para 60 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.	
			Execução agendada para 120 segundos			Execução agendada para 60 segundos	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 16							
ENTRADAS:							
Dados do usuário:	Usuario5, 29 anos, 70 kg, homem, 68 bpm, 18 ipm, 110/80 mmHg, 99% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>		
FC	70	70	68	67	65		
FR	18	18	18	18	17		
PAS	125	125	120	115	115		
PAD	82	82	80	80	80		
OS	98	98	100	100	100		
TC	37	37	37	37	37		
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity		
Base de regras	3, 10, 11, 12, 13, 14, 17, 18, 19						
SAÍDAS:	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado		
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.		
	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos		
				Execução agendada para 120 segundos			

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 17							
ENTRADAS:							
Dados do usuário:	Usuario6, 50 anos, 70 kg, mulher, 95 bpm, 17 ipm, 120/85 mmHg, 97% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>			
FC	110	115	100	98			
FR	18	18	18	16			
PAS	135	140	130	120			
PAD	85	90	90	85			
OS	97	97	96	96			
TC	36	36	36	36			
Estado	Sleep	Stand	Stand	Stand			
Base de regras	3, 10, 11, 12, 13, 14, 17, 18, 19						
SAÍDAS:	Nenhuma regra acionada	Nenhuma regra acionada	Nenhuma regra acionada	Ativação da regra 14: Normal detectado			
	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Medição salva no banco de dados local.	Ativação da regra 18: Normal detectado			
	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Medição e regra acionada salvas no banco de dados local.			
				Execução agendada para 120 segundos			

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 18							
ENTRADAS:							
Dados do usuário:	Usuario7, 40 anos, 63 kg, mulher, 95 bpm, 17 ipm, 110/75 mmHg, 97% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	
FC	90	90	95	95	95	100	
FR	17	17	16	16	18	18	
PAS	110	110	115	120	120	120	
PAD	78	80	80	83	85	85	
OS	96	96	96	96	97	97	
TC	37	37	37	37	37	37	
Estado	Stand	Stand	Stand	Stand	Stand	Stand	
Base de regras	3, 10, 11, 12, 13, 14, 17, 18, 19						
SAÍDAS:	Ativação da regra 14: Normal detectado	Ativação da regra 14: Normal detectado	Ativação da regra 14: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	
	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Ativação da regra 18: Normal detectado	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	
	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Medição e regra acionada salvas no banco de dados local.	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Execução agendada para 120 segundos	
	Execução agendada para 120 segundos	Execução agendada para 120 segundos	Envio da base de dados local para o centro de saúde e limpeza da base local.				
			Execução agendada para 120 segundos				

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 19							
ENTRADAS:							
Dados do usuário:	Usuario8, 53 anos, 70 kg, homem, 63 bpm, 14 ipm, 120/80 mmHg, 99% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	58	60	60	65	51	62	60
FR	14	15	14	15	15	14	14
PAS	110	120	125	130	127	125	120
PAD	75	78	80	85	82	80	80
OS	94	97	99	98	98	100	100
TC	36	37	37	37	37	37	37
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	Stand	Stand	Stand
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 12: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 15: Normal detectado.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 14: Normal detectado.	Ativação da regra 14: Normal detectado.
	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.
	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 60 segundos.		Execução agendada para 60 segundos.	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.
						Execução agendada para 120 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 20							
ENTRADAS:							
Dados do usuário:	Usuario9, 65 anos, 65 kg, mulher, 90 bpm, 15 ipm, 120/70 mmHg, 99% oxigenação, 36 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	90	100	120	100	160	120	90
FR	15	16	17	16	18	17	15
PAS	120	120	130	120	120	130	120
PAD	70	70	80	70	80	80	80
OS	95	100	98	98	100	98	97
TC	37	37	36	36	36	36	36
Estado	Sleep	Sleep	Stand	Stand	Stand	HomeActivity	HomeActivity
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 13: Normal detectado.	Ativação da regra 13: Normal detectado.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 15: Normal detectado.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.
	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Medição e regras acionadas salvas no banco de dados local.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.			Execução agendada para 60 segundos.	Execução agendada para 60 segundos.
			Execução agendada para 60 segundos.				

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 21							
ENTRADAS:							
Dados do usuário	Usuario10, 37 anos, 90 kg, mulher, 80 bpm, 18 ipm, 100/80 mmHg, 92% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	80	80	82	84	85	85	80
FR	16	17	17	18	18	17	17
PAS	110	115	120	125	125	120	120
PAD	75	77	80	80	85	80	80
OS	90	90	90	93	91	91	92
TC	36	36	37	37	37	37	37
Estado	HomeActivity	HomeActivity	Stand	Sleep	Sleep	Sleep	Sleep
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 15: Normal detectado.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 2: Emergência detectada; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 2: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.
	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Envio da base de dados local para o centro de saúde e limpeza da base local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 30 segundos.	Execução agendada para 30 segundos.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
			Execução agendada para 60 segundos.			Execução agendada para 60 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 22							
ENTRADAS:							
Dados do usuário:	Usuario8, 53 anos, 70 kg, homem, 63 bpm, 14 ipm, 120/80 mmHg, 99% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	98	102	90	90	72	62	60
FR	14	15	14	15	15	14	14
PAS	150	145	130	130	127	125	120
PAD	90	85	85	85	82	80	80
OS	94	97	99	98	98	100	100
TC	36	37	37	37	37	37	37
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.
	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 17: Alerta detectado; Mensagem enviada ao centro de saúde.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.	Execução agendada para 120 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.
			Execução agendada para 120 segundos.			Execução agendada para 120 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 23							
ENTRADAS:							
Dados do usuário:	Usuario9, 65 anos, 65 kg, mulher, 90 bpm, 15 ipm, 120/70 mmHg, 99% oxigenação, 36 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	90	100	120	100	160	120	90
FR	15	16	17	16	18	17	15
PAS	120	120	130	140	145	130	120
PAD	70	70	80	70	80	80	80
OS	95	100	98	98	100	98	97
TC	37	37	36	36	36	36	36
Estado	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 13: Normal detectado.	Ativação da regra 13: Normal detectado.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 2: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 13: Normal detectado.
	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.	Execução agendada para 30 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.
			Execução agendada para 60 segundos.			Execução agendada para 60 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 24							
ENTRADAS:							
Dados do usuário:	Usuario10, 37 anos, 90 kg, mulher, 80 bpm, 18 ipm, 100/80 mmHg, 92% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	95	99	99	100	85	85	80
FR	16	17	17	18	18	17	17
PAS	115	120	125	130	125	120	120
PAD	75	77	80	85	85	80	80
OS	90	90	90	93	91	91	92
TC	36	36	37	37	37	37	37
Estado	Stand	Stand	Stand	Stand	Stand	Stand	Stand
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 3: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 3: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 3: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 8: Alerta detectado; Mensagem enviada ao centro de saúde.
	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 30 segundos.	Execução agendada para 30 segundos.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.
			Execução agendada para 30 segundos.				

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

[illegible]

	detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	banco de dados local.	banco de dados local.	banco de dados local.	banco de dados local.	banco de dados local.	banco de dados local.
	Medição e regras acionadas salvas no banco de dados local.	Execução agendada para 30 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.
	Execução agendada para 30 segundos.		Execução agendada para 60 segundos.			Execução agendada para 120 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 26							
ENTRADAS:							
Dados do usuário:	Usuario9, 65 anos, 65 kg, mulher, 90 bpm, 15 ipm, 120/70 mmHg, 99% oxigenação, 36 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	90	100	120	100	160	120	90
FR	15	16	17	16	18	17	15
PAS	120	120	130	140	145	130	120
PAD	70	70	80	70	80	80	80
OS	95	100	98	98	100	98	97
TC	37	37	36	36	36	36	36
Estado	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity	HomeActivity
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 9: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 15: Normal detectado.	Ativação da regra 15: Normal detectado.
	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 120 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 120 segundos.
			Execução agendada para 60 segundos.			Execução agendada para 60 segundos.	

continua

Tabela C.2 – Casos de teste elaborados para verificar o comportamento do MonitorPrevIntel.

continuação

Caso de Teste 27							
ENTRADAS:							
Dados do usuário:	Usuario10, 37 anos, 90 kg, mulher, 80 bpm, 18 ipm, 100/80 mmHg, 92% oxigenação, 37 °C						
	<i>Medição 1</i>	<i>Medição 2</i>	<i>Medição 3</i>	<i>Medição 4</i>	<i>Medição 5</i>	<i>Medição 6</i>	<i>Medição 7</i>
FC	95	99	99	100	85	85	80
FR	16	17	17	18	18	17	17
PAS	115	120	125	130	125	120	120
PAD	75	77	80	85	85	80	80
OS	90	90	90	93	91	91	92
TC	36	36	37	37	37	37	37
Estado	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep
Base de regras	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21						
SAÍDAS:	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 2: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 2: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 2: Emergência detectado; Mensagem enviada ao centro de saúde; Sugestão ao paciente para executar uma ligação ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 7: Alerta detectado; Mensagem enviada ao centro de saúde.
	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 16: Alerta detectado; Mensagem enviada ao centro de saúde.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.	Ativação da regra 18: Normal detectado.
	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.	Medição e regras acionadas salvas no banco de dados local.
	Execução agendada para 60 segundos.	Execução agendada para 60 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 30 segundos.	Execução agendada para 30 segundos.	Envio da base de dados local para o centro de saúde e limpeza da base local.	Execução agendada para 60 segundos.
			Execução agendada para 30 segundos.			Execução agendada para 60 segundos.	

Anexo A – Regras de Produção Utilizadas para Teste do MonitorPrevIntel

Regras de produção utilizadas pelo sistema de produção para teste do aplicativo *Monitor Intelligent Prevention*, este conjunto de regras foi retirado da Tese de COPETTI (2010), p. 107-109.

if (atividade is dormir) and (desvioPASdormindo is muitoBaixo) then (acao is emergencia) (1)

if (atividade is dormir) and (desvioPASdormindo is baixo) then (acao is alerta) (0.3)

if (atividade is dormir) and (desvioPASdormindo is normal) then (acao is normal) (0.2)

if (atividade is dormir) and (desvioPASdormindo is alto) then (acao is alerta) (0.5)

if (atividade is dormir) and (desvioPASdormindo is muitoAlto) then (acao is emergencia) (1)

if (atividade is repouso) and (desvioPASrepouso is muitoBaixo) then (acao is emergencia) (1)

if (atividade is repouso) and (desvioPASrepouso is baixo) then (acao is alerta) (0.3)

if (atividade is repouso) and (desvioPASrepouso is normal) then (acao is normal) (0.2)

if (atividade is repouso) and (desvioPASrepouso is alto) then (acao is alerta) (0.5)

if (atividade is repouso) and (desvioPASrepouso is muitoAlto) then (acao is emergencia) (1)

if (atividade is ativDomestica) and (desvioPASativDomestica is muitoBaixo) then (acao is emergencia) (1)

if (atividade is ativDomestica) and (desvioPASativDomestica is baixo) then (acao is alerta) (0.3)

if (atividade is ativDomestica) and (desvioPASativDomestica is normal) then (acao is normal) (0.2)

if (atividade is ativDomestica) and (desvioPASativDomestica is alto) then (acao is alerta) (0.5)

if (atividade is ativDomestica) and (desvioPASativDomestica is muitoAlto) then (acao is emergencia) (1)

if (freqCard is normal) then (acao is normal) (0.1)

if (freqCard is baixa) then (acao is alerta) (0.1)

if (freqCard is alta) then (acao is alerta) (0.1)

if (atividade is dormir) and (freqCard is alta) then (acao is alerta) (0.7)

if (atividade is not dormir) and (desvioPASrepouso is muitoAlto) and (PAS is hipertenso) then (acao is emergencia) (1)

if (atividade is not dormir) and (desvioPADrepouso is muitoAlto) and (PAD is hipertenso) then (acao is emergencia) (1)