UNIVERSIDADE CATÓLICA DE PELOTAS
CENTRO POLITÉCNICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

uMED: Uma Arquitetura para Desenvolvimento de Software Direcionada à Medicina Ubíqua

por
Sérgio Luis Rodrigues

Dissertação apresentada como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Orientador: Prof. Dr. Adenauer Corrêa Yamin

DM-2010-2-009

Pelotas, março de 2011

AGRADECIMENTOS

Agradeço a Deus e a minha família pois sem eles nada seria possível.

Agradeço a meu Orientador Prof. Dr. Adenauer Corrêa Yamin, por sua dedicação, amizade e por sua orientação.

Ao Instituto Federal Sul-rio-grandense pelo apoio e incentivo nesta experiência tão gratificante, especial e renovadora, a realização do meu Curso de Mestrado em Ciência da Computação.

Agradeço aos integrantes do G3PD, Grupo de Processamento Paralelo Distribuído, em especial à Amanda, pela colaboração e apoio no desenvolvimento das aplicações empregadas no estudo de caso.

Agradeço aos colegas do PPGINF que me receberam muito bem e caminham comigo neste percurso. Aos funcionários e professores do Centro Politécnico da Universidade Católica de Pelotas pela seriedade e compromisso.

Por último, agradeço a todos aqueles que não foram mencionados, mas que de alguma forma também contribuíram para a elaboração deste trabalho.

A vida é o sopro do SUPREMO
CRIADOR o GRANDE ARQUITETO
DO UNIVERSO numa atitude
repleta de amor...!
— GONZAGUINHA

SUMÁRIO

LISTA DE FIGURAS 7

LISTA DE TABELAS 9

LISTA DE ABREVIATURAS E SIGLAS 10

RESUMO 13

ABSTRACT 14

1 INTRODUÇÃO 15
1.1 Tema 17
1.2 Contexto de Pesquisa 17
1.2.1 Projeto PERTMED 17
1.2.2 Middleware EXEHDA 17
1.3 Motivação 18
1.4 Objetivos 20
1.5 Estrutura do texto 20

2 COMPUTAÇÃO UBÍQUA 22
2.1 Características da Computação Ubíqua 24
2.2 Cenários em Computação Ubíqua 25
2.3 Tendências em Computação Ubíqua 27
2.4 Considerações sobre o Capítulo 28

3 MEDICINA UBÍQUA: IDENTIFICANDO CARACTERÍSTICAS E TECNOLOGIAS 29
3.1 O Ambiente de Medicina Ubíqua 30
3.1.1 Hospital do Futuro: Idéias e Conceitos 31
3.1.2 Dispositivos Interativos em Ambientes Hospitalares 32
3.1.3 Contribuições para Área da Saúde 32
3.2 Telemedicina e Medicina Ubíqua: Revisando Conceitos 33
3.3 Medicina Ubíqua: Principais Aspectos de Estudo e Pesquisa 34
3.3.1 Caso 1 - Suporte à Mobilidade 35
3.3.2 Caso 2 - Suporte à Heterogeneidade 36
3.3.3 Caso 3 - Suporte à Ética Médica 37
3.4 Considerações sobre o Capítulo 38

4 PROJETOS DE MEDICINA UBÍQUA: SISTEMATIZANDO SUAS CARACTERÍSTICAS 39
4.1 Projeto ABC 39
4.2 Projeto Awareness 42
4.3 Projeto UbiDoctor 44
4.4 Projeto ClinicSpace 47
4.5 Síntese dos Trabalhos Relacionados 50
4.6 Considerações sobre o Capítulo 52

5 UMED: FUNDAMENTOS 53
5.1 Projeto PERTMED 53
5.2 Middleware EXEHDA 56
5.2.1 Aspectos Funcionais e Arquiteturais 56
5.3 O Serviço EXEHDA-SS 63
5.3.1 Concepção e Modelagem do EXEHDA-SS 64
5.3.2 Arquitetura de Software do EXEHDA-SS 65
5.4 Considerações sobre o Capítulo 66

6 UMED: CONCEPÇÃO E MODELAGEM 67
6.1 Arquitetura de Software do uMED 67
6.1.1 Gerente de Atuação 69
6.1.2 Gerente de Aplicações 71
6.1.3 Gerente de Borda 74
6.1.4 Gerente de Comunicação 78
6.1.5 Servidor de Contexto 80
6.2 Considerações sobre o Capítulo 83

7 UMED: TECNOLOGIAS UTILIZADAS E ESTUDO DE CASO 84
7.1 Principais Tecnologias Utilizadas 84
7.1.1 Java 84
7.1.2 JSF 85
7.1.3 PostgreSQL 85
7.1.4 Web Service 85
7.1.5 XML 86
7.1.6 DB4O 86
7.1.7 YAML 86
7.2 Estudo de Caso - Acompanhamento Clínico de Pacientes 87
7.2.1 Objetivos da ACP 87
7.2.2 Monitoramento Pró-ativo de Pacientes 88
7.2.3 Coleta Pró-ativa de Sinais Vitais 89
7.2.4 Geração de Relatórios 89
7.2.5 Atuação Pró-ativa de Pacientes 91
7.2.6 Configuração de Envio de Notificações 91
7.3 Considerações sobre o Capítulo 93

8 CONSIDERAÇÕES FINAIS 94
8.1 Principais Contribuições 95
8.2 Publicações Realizadas 95
8.3 Trabalhos Futuros 97

REFERÊNCIAS 99

ANEXO A PADRÕES DE SENSORES E ATUADORES 104

ANEXO B CARACTERIZANDO SINAIS VITAIS DE PACIENTES . 106

LISTA DE FIGURAS

Figura 2.1 Computação Ubíqua (SAHA; MUKHERJEE, 2003) 23
Figura 2.2 Framework para Computação Ubíqua (adaptada de (SAHA; MUKHERJEE, 2003)) 23

Figura 3.1 Dispositivos Interativos (BARDRAM; CHRISTENSEN, 2007) 32

Figura 4.1 O Núcleo do Middleware ABC (PEDERSEN; MOGENSEN; BARDRAM, 2006) 40
Figura 4.2 Arquitetura do Projeto ABC (BARDRAM; CHRISTENSEN, 2007) . 41
Figura 4.3 Camadas do Awareness (adaptada de (WEGDAM, 2005)) 43
Figura 4.4 Arquitetura do Protótipo UHSys (DINIZ, 2009) 45
Figura 4.5 Arquitetura de Serviços UbiDoctor (DINIZ, 2009) 46
Figura 4.6 Protótipo do ClinicSpace (SILVA, 2010) 48
Figura 4.7 Arquitetura para Programação e Gerenciamento de Tarefas (SILVA, 2010) 48

Figura 5.1 Arquitetura de Software do Middleware EXEHDA (YAMIN, 2004) . 59
Figura 5.2 Ambiente Ubíquo Suprido pelo EXEHDA (YAMIN, 2004) 60
Figura 5.3 Organização dos Subsistemas do EXEHDA (YAMIN, 2004) 61
Figura 5.4 Organização do Núcleo do EXEHDA (YAMIN, 2004) 62
Figura 5.5 Classes da OntContext (VENECIAN, 2010) 65

Figura 6.1 Arquitetura de Software Proposta para uMED 68
Figura 6.2 Fluxo de Informações para o Gerente de Borda 72
Figura 6.3 Exemplo de Arquivo `SensorConfiguracao` 75
Figura 6.4 Exemplo de Arquivo `AtuadorConfiguracao` 76
Figura 6.5 Vista Expandida do Gerente de Borda 76
Figura 6.6 Diagrama de Sequência de Mensagem Off-line Disparada Manualmente 79
Figura 6.7 Diagrama de Sequência de Mensagem On-line Disparada pelo Contexto 79
Figura 6.8 Fluxo do Motor de Inferência do EXEHDA-SS (VENECIAN, 2010) . 81
Figura 6.9 Exemplo de Notificação às Aplicações (VENECIAN, 2010) 82

Figura 7.1 Bancos Objeto-Relacional e Banco de Objetos db4o (DB4O, 2010) . 87
Figura 7.2 Monitoramento Pró-ativo de Pacientes 89
Figura 7.3 Configuração dos Sensores 90
Figura 7.4 Relatório de Monitoramento por Faixa de Valores do Dispositivo . . . 90
Figura 7.5 Relatório de Atuação por Período de Tempo 91
Figura 7.6 Relatório de Histórico Clínico 92

Figura 7.7 Configuração dos Atuadores 92
Figura 7.8 Configuração de Envio de Notificações 93

LISTA DE TABELAS

Tabela 4.1 Síntese dos Trabalhos Relacionados 52

Tabela 6.1 Parâmetros Operacionais dos Atuadores 70
Tabela 6.2 Parâmetros Operacionais dos Sensores 75

Tabela 8.1 Principais Contribuições do uMED 96

LISTA DE ABREVIATURAS E SIGLAS

ABC	Activity-Based Computing
ACP	Acompanhamento Clínico de Pacientes
ATA	American Telemedicine Association
AVU	Ambiente Virtual do Usuário
BAN	Body Area Network
BDA	Base de Dados Ubíqua das Aplicações
BSD	Berkeley Software Distribution
CFM	Conselho Federal de Medicina
CIB	Cell Information Base
CO2	Gás Carbônico
DC	Dynamic Configurator
EXEHDA	Execution Environrnent for High Distributed Applications
GMob	Grupo de Sistemas em Computação Móvel
Gtalk	Google Talk
GPRS	General Packet Radio Service
GPS	Global Positioning System
G3PD	Grupo de Pesquisa em Processamento Paralelo e Distribuído
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ISAM	Infraestrutura de Suporte à Aplicações Móveis
JME	Java Micro Edition
JSF	JavaServer Faces
JSP	Java Server Pages
LED	Díodo Emissor de Luz

NASA	National Air and Space Agency
MmHg	Milímetro de Mercúrio
OMS	Organização Mundial de Saúde
OWL	Web Ontology Language
OX	Objeto eXehda
O2	Oxigênio
PEP	Prontuário Eletrônico do Paciente
P2P	Peer-to-Peer
PPGINF	Programa de Pós-graduação em Informática
PDA	Personal Digital Assistant
RA	Repositório de Aplicações
RC	Repositório de Comunicações
RCN	Repositório de Contextos Notificados
RFC	Request for Comments
RIC	Repositório de Informações Contextuais
RTCA	Repositório Temporário de Comandos de Atuação
RTDC	Repositório Temporário de Dados Coletados
SGBDOR	Sistema Gerenciador de Banco de Dados Objeto-relacional
SGBDs	Sistema Gerenciador de Banco de Dados
SGDT	Subsistema de Gerenciamento Distribuído de Tarefas
SGML	Standard Generalized Markup Language
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SpO2	Saturação do Oxigênio
UbiComp	Computação Ubíqua
UCPeL	Universidade Católica de Pelotas
UFPeL	Universidade Federal de Pelotas
UFRGS	Universidade Federal do Rio Grande do Sul
UFSM	Universidade Federal de Santa Maria
uMED	ubiquitons MEDicine
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UTI	Unidade de Tratamento Intensivo

USB	Universal Serial Bus
UHS	Ubiquitous Health Service
UHSys	Ubiquitous Health System
WEB	World Wide Web
WSDL	Web Services Description Language
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

RESUMO

A medicina ubíqua tem como premissa disponibilizar acesso aos serviços de saúde a qualquer hora, sem restrições de localização potencializando a mobilidade de médicos, enfermeiros e outros profissionais de saúde. Estes profissionais necessitam de ferramentas de acesso às informações, tanto no local onde se encontra o paciente, como não. O objetivo central deste trabalho é propor uma infraestrutura que integre sensores e dispositivos computacionais (móveis e fixos), valendo-se para isto do ambiente ubíquo provido por um *middleware*, e de uma arquitetura de *software* com sensibilidade para elementos de contexto, e também com capacidade de atuação sobre os mesmos.

Esta arquitetura é denominada uMED, e suas respectivas aplicações alvo irão contemplar o monitoramento de sinais vitais de pacientes considerando um ambiente típico de trabalho das equipes que atuam em urgências médicas. Também está sendo prevista a possibilidade dos profissionais de saúde controlarem remotamente (atuação) equipamentos médicos. A partir das funcionalidades de monitoramento e/ou atuação pretende-se possibilitar uma melhora nas rotinas de trabalho dos profissionais de saúde, contribuindo para reduzir o elevado percentual de nomadismo e fragmentação de suas atividades. A arquitetura do uMED está sendo concebida com a perspectiva de ser integrada ao *middleware* EXEHDA.

No sentido de avaliar a arquitetura de software proposta para o uMED, foi desenvolvido um estudo de caso direcionado ao acompanhamento clínico de pacientes.

Palavras-chave: Medicina Ubíqua, Computação Ubíqua, Sensibilidade ao Contexto, Aplicações em Saúde.

TITLE: "UMED: AN ARCHITECTURE FOR SOFTWARE DEVELOPMENT TARGETED TO UBIQUITOUS MEDICINE"

ABSTRACT

The ubiquitous medicine has the premisse to provide access to health services at any time, without restriction of location enhancing the mobility of doctors, nurses and other health professionals. These professionals need tools for information access, both at the site where the patient, either in other locations. The aim of this paper is to propose an infrastructure that integrates sensors and computing devices (mobile and fixed), considering the ubiquitous environment provided by a middleware and software architecture with context elements awareness, and also with the capacity acting on them.

This architecture is called uMED, and their target applications will include monitoring vital signs of patients considering a typical work environment of staff working in medical emergencies. It is also being made to allow health professionals to remotely control (actuation) medical equipments. From the monitoring and/or actuation features, the intention is to enable an improvement in work routines of health professionals, helping to reduce the high percentage of nomadism and fragmentation of its activities. The uMED architecture is being designed with the perspective of being integrated with middleware EXEHDA.

In order to evaluate the software architecture proposed for uMED, was developed a case study aimed at clinical follow-up of patients.

Keywords: Ubiquitous Medicine, Ubiquitous Computing, Context-Aware, Health Applications.

1 INTRODUÇÃO

As tecnologias mais profundas são aquelas que desaparecem. Elas se itegram no tecido da vida cotidiana, até que são indistinguíveis a partir dele. Resume o visionário "Mark Weiser" (WEISER, 1991) quando da declaração do que se espera da computação pervasiva ou ubíqua: o acesso do usuário ao ambiente computacional, em todos os lugares, todo tempo, por meio de qualquer dispositivo.

Na proposta da computação ubíqua (COSTA; YAMIN; GEYER, 2008) o processamento está distribuído no ambiente através de vários dispositivos, que realizam tarefas bem definidas dependendo de sua natureza, interligados de forma que essa infraestrutura torna-se o mais imperceptível possível para o usuário. Aplicações ubíquas executam em ambientes instrumentados com sensores[1], e atuadores[2] geralmente dotados de interfaces de redes sem fio, nos quais dispositivos, agentes de *software* e serviços são integrados de forma transparente e cooperam para atender aos objetivos da aplicação. Essa categoria de aplicações caracteriza-se por constantes mudanças em seu contexto de execução, geradas pelos ambientes altamente dinâmicos em que tipicamente acontece seu processamento.

A computação ubíqua vem conquistando espaços com a disseminação de dispositivos portáteis, sobretudo após a ampliação das tecnologias de redes sem fio. Dessa forma, o uso desta computação tem conduzido a aplicações de diversas áreas do mundo real, onde sua usabilidade nos moldes da computação tradicional não poderia ser tão ampla. Cenários na área médica, podem estender-se a diversas atividades, como telemonitoramento através do uso de sensores de medicina ubíqua, diagnósticos remotos, segunda opinião, juntas médicas, *Home Care* viabilizando a integração de dispositivos móveis como PDAs (*Personal Digital Assistants*), *Tablets, Smartphones* com os sistemas de prontuário eletrônico existentes nos hospitais, entre outros cenários, desde residências inteligentes a trabalhos colaborativos (DINIZ, 2009).

Segundo Diniz (DINIZ, 2009) ambientes de medicina ubíqua são aqueles em que facilidades tecnológicas, como dispositivos móveis e redes de comunicação sem fio, trazem novas possibilidades de acesso e interação de seus usuários, como por exemplo, o acesso das informações dos pacientes. Estas informações compõem o PEP (Prontuário Eletrônico do Paciente), permitindo que dados sobre exames, fatos e situações sobre a saúde de um paciente possam ser acessados através de múltiplos dispositivos e redes heterogêneas. Na medicina ubíqua podem-se realizar acessos a PEPs com informações con-

[1] Sensores são dispositivos que detectam um estímulo físico ou químico (ex. temperatura, luminosidade, umidade, entre outras) e emite um impulso mensurável correspondente.

[2] Atuadores são dispositivos que agem quando estimulados, promovendo ações que alteram características físicas ou químicas no meio onde está inserido (ex. acionamento de uma campainha, acionamento de umidificador de ar, entre outras).

solidadas sobre os pacientes de qualquer lugar da rede, permitindo, inclusive, que haja cooperação entre profissionais independentemente do tempo e do espaço.

O PEP surgiu com a resolução do CFM (Conselho Federal de Medicina) Nº 1.821, de 11 de julho de 2007, que aprovou as normas técnicas concernentes à digitalização e uso dos sistemas informatizados para a guarda e manuseio dos documentos dos prontuários dos pacientes, autorizando a eliminação do papel e a troca de informação identificada em saúde (CFM, 2007).

Como uma das motivações para medicina ubíqua, temos a necessidade que ambientes computacionais ofereçam suporte à mobilidade dos profissionais de saúde, tendo em vista que a mobilidade dos médicos é inerente à própria profissão. Além desse caráter nômade do médico, é importante considerar que a atividade médica é bastante fragmentada (TENTORI; FAVELA, 2008), ou seja, está sujeita a interrupções durante sua execução, uma vez que médicos passam pouco tempo em cada local ou atividade. Dessa forma, mecanismos que facilitem a continuidade de atividades dos profissionais, mesmo em virtude de seus constantes deslocamentos, tendem a melhorar a produtividade dos mesmos (DINIZ, 2009).

Segundo Rodriguez (RODRIGUEZ, 2004) os médicos e colaboradores de hospitais trabalham em constante movimento. O trabalho descrito em (RODRIGUEZ, 2004) também se utiliza desta motivação reafirmando a necessidade de constante mudança de localização por parte destes profissionais em suas atividades diárias.

Ao se construir e executar aplicações para medicina ubíqua, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir do conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, armazenamento, e a realização de inferências para seu uso em tomadas de decisão. Em vez dessas funcionalidades ficarem a cargo da aplicação, e deste modo incorporandas ao código do negócio, são utilizadas plataformas ou *middlewares* de provisão de contexto (DINIZ, 2009).

Esse trabalho está inserido nos esforços de pesquisa do projeto PERTMED (PERTMED, 2007) (Sistema de TeleMedicina Móvel), a principal razão desta inserção é o fato da área médica, estar susceptível aos avanços das tecnologias móveis e sem fio, como *Bluetooth*, WiFi, GPRS (*General Packet Radio Service*) , os quais somados a popularização dos dispositivos móveis e sem fio, *Smartphone*, PDA, GPS (*Global Positioning System*) e pequenos dispositivos médicos, como *Holters*, entre outros, facilitam a tarefa de monitoramento de pacientes. Infraestruturas de *software* para o gerenciamento dessas informações contextuais necessitam, em geral, coletar uma grande quantidade de informações de diferentes naturezas do ambiente, analisando essas informações como variáveis independentes, ou combiná-las com outras informações do passado ou presente. Além disso, essas aplicações são caracterizadas por apresentarem contextos altamente dinâmicos e variados, com um grande grau de mobilidade dos seus principais atores (médicos, pacientes, técnicos, entre outros).

Na perspectiva de suprir estas funcionalidades, este trabalho tem como objetivo principal propor um *framework* denominado uMED (ubiquitons MEDicine) direcionado para demandas da medicina ubíqua.

1.1 Tema

O trabalho proposto tem como tema o desenvolvimento de uma arquitetura de *software* na perspectiva da computação ubíqua direcionada para área da medicina. Os ambientes de medicina ubíqua são aqueles em que facilidades tecnológicas, como dispositivos móveis e redes de comunicação sem fio, trazem novas possibilidades de acesso e interação de seus usuários (médicos, enfermeiros, pacientes, dentre outros) tornando o serviço mais ágil e abrangendo regiões carentes de serviços médicos.

Outrossim o tema deste trabalho também abrange estudar projetos já existentes na área de medicina ubíqua assim como as soluções utilizadas para solucionar os desafios impostos nesta área. A pesquisa irá culminar com a concepção de um *framework* denominado uMED, no qual visa atender demandas da medicina ubíqua.

1.2 Contexto de Pesquisa

O contexto de pesquisa desta dissertação é composto por dois projetos que serão resumidos a seguir.

1.2.1 Projeto PERTMED

Devido aos constantes avanços tecnológicos o sistema de saúde será favorecido com uso de tecnologias como a da computação ubíqua onde é formado um espaço inteligente (reativo e pró-ativo), onde dispositivos móveis e fixos estão integrados ao ambiente físico (objetos) visando captar informações do meio e transmitir as alterações para sistemas de gerenciamento de informações, os quais tomarão decisões e adaptar-se-ão às situações detectadas, segunda as premissas da computação consciente de contexto.

O projeto PERTMED (PERTMED, 2007) propõe fazer a ligação entre os sistemas automatizados existentes (registro de pacientes, exames laboratoriais, entre outros) e o médico no local em que este se encontra (regiões remotas ou em trânsito, por exemplo). Desta forma, elimina-se a exigência de uma conexão a uma rede fixa ou com um computador pessoal na área do hospital para ter acesso às informações do paciente.

O projeto PERTMED prevê o uso do EXEHDA (*Execution Environrnent for High Distributed Applications*) como *middleware* direcionado a computação ubíqua. Neste sentido, este trabalho busca atender as demandas do projeto PERTMED através da utilização dos serviços do *middleware* EXEHDA para o desenvolvimento de uma arquitetura para medicina ubíqua denominada uMED.

1.2.2 Middleware EXEHDA

Segundo Yamin (YAMIN, 2004) o EXEHDA é um *middleware* adaptativo ao contexto e baseado em serviços que visa criar e gerenciar um ambiente ubíquo, bem como promover a execução, sob este ambiente, das aplicações que expressam a semântica sigame. Estas aplicações são distribuídas, móveis e adaptativas ao contexto em que seu processamento ocorre, estando disponíveis a todo momento e em qualquer lugar.

O *middleware* EXEHDA faz parte dos esforços de pesquisa do Projeto ISAM (Infraestrutura de Suporte à Aplicações Móveis). O ISAM vem sendo desenvolvido por um consórcio de universidades do sul do Brasil, e foi iniciado na UFRGS (Universidade Federal do Rio Grande do Sul) sob a coordenação do Prof. Cláudio Geyer (ISAM, 2009).

Afim de prover a elevada flutuação na disponibilidade dos recursos, inerente à computação ubíqua, o EXEHDA é estruturado em um núcleo mínimo e mais os serviços carregados sob demanda. Os principais serviços fornecidos estão organizados em subsistemas que gerenciam: a execução distribuída, a comunicação, o reconhecimento do contexto, a adaptação, o acesso ubíquo aos recursos e serviços, a descoberta de recursos e o gerenciamento de recursos.

Conforme Yamin (YAMIN, 2004) no EXEHDA, as condições de contexto são pró-ativamente monitoradas e o suporte à execução deve permitir que tanto a aplicação como ele próprio utilizem essas informações na gerência da adaptação de seus aspectos funcionais e não-funcionais. O mecanismo de adaptação do EXEHDA emprega uma estratégia colaborativa entre aplicação e ambiente de execução, através da qual é permitido ao programador individualizar políticas de adaptação para reger o comportamento de cada um dos componentes que constituem o *software* da aplicação.

1.3 Motivação

É crescente o uso de Tecnologia de Informação na prestação de serviços de saúde. As atividades de segunda opinião, monitoramento de pacientes, telediagnóstico remoto, entre outras, são atividades desenvolvidas mediante uma infraestrutura de rede, realizadas por profissionais conectados em outro centro médico, via de regra distante da localidade onde o paciente está sendo examinado. Estes cenários tem se mostrado interessantes pois reduzem custos com relação ao deslocamento dos profissionais de saúde e com relação à troca de informações sobre diagnósticos. Todo o histórico de um paciente pode ser armazenado em um PEP, e através de um acesso remoto aos dados é possível o emprego do mesmo como um auxílio na obtenção de melhores diagnósticos.

De acordo com o Conselho Federal de Medicina, o prontuário do paciente é um documento único constituído de um conjunto de informações, sinais e imagens registradas, geradas a partir de fatos, acontecimentos e situações sobre a saúde do paciente e a assistência a ele prestada, de caráter legal, sigiloso e científico, que possibilita a comunicação entre membros da equipe multiprofissional e a continuidade da assistência prestada ao indivíduo (CFM, 2002).

Com a expansão das redes de comunicação, possibilitando larguras de banda superiores, comunicação sem fio e acesso através de dispositivos de menores dimensões, como *Notebooks, Smartphones*, PDAs, entre outros, o uso de tecnologia móvel nos ambientes de saúde tornou-se um desafio possível. Através do uso de dispositivos de menores dimensões e de enlaces de comunicação sem fio e de alta velocidade, é possível se fazer acesso ao PEP, de maneira ubíqua, conforme conceito introduzido por Weiser e colegas (WEISER; GOLD; BROWN, 1999). Segundo Weiser, através da computação ubíqua, o usuário pode ter acesso ao seu ambiente computacional a partir de qualquer lugar, em qualquer momento, de várias formas, através de dispositivos diferentes e com as mais variadas formas de tecnologias de comunicação.

Weiser e colegas (WEISER; GOLD; BROWN, 1999) introduziram a visão que computadores portáteis com dimensões reduzidas estarão embutidos em objetos do cotidiano do usuário, tais como livros, roupas e objetos pessoais em geral, interagindo entre si e adaptando o comportamento de suas aplicações, de acordo com informações relevantes à sua execução. Dessa forma, estes elementos seriam sensíveis ao contexto que os cercam, em busca de informações que possam aprimorar a execução de suas aplicações.

Segundo Diniz (DINIZ, 2009) ambientes de prestação de serviços de saúde, o suporte à mobilidade é um fator indispensável, tendo em vista que a mobilidade dos médicos é inerente à própria profissão. Além deste caráter nômade do médico é importante considerar que a atividade médica é bastante fragmentada, ou seja, está sujeita a interrupções durante sua execução, pois poderá realizar consultas presenciais, realizar cirurgias, visitas a pacientes, em deslocamento para sua residência ou local de trabalho, de sobreaviso em sua casa ou local de lazer, e ser interrompido por um paciente ou colega para a tomada de alguma decisão com relação à saúde de algum paciente. Da mesma forma, o médico poderá realizar uma consulta ou inclusão de dados de algum prontuário eletrônico de paciente e ser interrompido para a realização de alguma atividade de urgência. Em cada uma dessas situações, o médico poderá ter à sua disposição diferentes dispositivos, tais como *Personal Computer* (PCs), *Notebooks, Tablets, Palmtops, Smartphones*, PDAs com configurações diferentes e poderá fazer acesso ao sistema de informações de saúde de qualquer um deles.

Para verificar o caráter nômade e fragmentado da atividade médica, destaca-se o trabalho descrito por Tentori e Favela (TENTORI; FAVELA, 2008). Segundo os autores eles realizaram um estudo de caso em um hospital público, onde observaram as práticas do corpo clínico dentro do hospital. Foi constatado que os funcionários da equipe pesquisada (médicos, enfermeiros, técnicos entre outros) se movem mais de 50% do tempo. A análise mostrou ainda que em 69% do tempo eles estão interagindo com outras pessoas e que 26% dessas interações envolvem dispositivos, como por exemplo, telefones. Os funcionários gastam em média 2,5 minutos em cada interação. Também conseguiram verificar que o trabalho desempenhado é altamente fragmentado. Os médicos não gastam mais que cinco minutos conduzindo uma atividade, sem serem interrompidos. Eles também mensuraram o tempo de transição entre atividades, ou seja, o tempo decorrido entre duas atividades consecutivas, e verificaram, então, que este tempo chegava, em média, a 51 segundos. Essa fragmentação, normalmente, ocorre devido a uma necessidade de interrupção, devido a um chamado ou alguma atividade de urgência.

Conforme pode ser observado, as demandas da medicina ubíqua são elevadas, a proposta desta dissertação visa atender algumas demandas que auxiliem os profissionais na área de saúde em sua rotina diária nos hospitais ou fora deles, sobretudo no relacionamento com os pacientes, no que diz respeito a acompanhamento médico à distância. Neste sentido, considera-se um ambiente de medicina ubíqua, aquele onde há diversos dispositivos computacionais (heterogêneos) nos vários locais por onde um profissional de saúde passa, e o *software* de auxílio ao trabalho médico seja capaz de perceber variações no estado do paciente e gerar alertas ou até mesmo promover a atuação de dispositivos médicos associados ao paciente.

O grupo G3PD (Grupo de Pesquisa em Processamento Paralelo e Distribuído) relacionado ao qual este trabalho está sendo desenvolvido vem ultimamente pesquisando sobre computação ubíqua/pervasiva desenvolvendo serviços para um *middleware* chamado EXEHDA. Neste trabalho são identificados os desafios de pesquisa na área de medicina ubíqua e com os resultados a serem obtidos a perspectiva é adequar o uso dos serviços do *middleware* EXEHDA para as demandas funcionais introduzidas pela medicina ubíqua.

1.4 Objetivos

O objetivo geral deste trabalho é explorar as premissas de estudo e desenvolvimento praticadas atualmente em medicina ubíqua, sistematizando os desafios de pesquisa existentes no estado da arte em UbiComp (Computação Ubíqua) aplicada a área médica. A partir desta sistematização propor um *framework* denominado uMED.

Como objetivos específicos desta dissertação, destacaríamos:

- sistematizar fundamentos teóricos sobre computação ubíqua;
- revisar fundamentos teóricos sobre medicina ubíqua, identificando as tecnologias relacionadas;
- identificar os principais projetos em medicina ubíqua, sistematizando suas diferentes características;
- avaliar o *middleware* EXEHDA, estudando os serviços a serem utilizados no uMED;
- propor um *framework* que atenda as demandas previstas para o uMED;
- difundir o conhecimento pertinente à área de medicina ubíqua, sobretudo no que diz respeito aos aspectos de aplicações sensíveis ao contextos;
- perseguir a integração com grupos que trabalham com medicina ubíqua no cenário nacional;
- fornecer subsídios para a elaboração de relatórios, artigos e trabalhos futuros, relacionados com o tema pesquisado.

A discussão destes objetivos, bem como os conceitos, as metodologias e técnicas para sua concepção estão registrados no texto cuja estrutura é apresentada a seguir.

1.5 Estrutura do texto

A estrutura do texto deste trabalho contempla oito capítulos e dois anexos.

No capítulo 1, que é esta introdução, é abordado o tema do trabalho, a motivação para escolha do mesmo e os objetivos propostos para o desenvolvimento deste trabalho.

No capítulo 2 são apresentados conceitos sobre computação ubíqua, discutindo as características relevantes.

O capítulo 3 faz um estudo detalhado sobre medicina ubíqua, suas características e tecnologias empregadas, assim como ambientes de medicina ubíqua.

No capítulo 4 discute projetos relacionados com o modelo proposto avaliando suas características, ao final deste capítulo e realizado uma síntese entre os mesmos.

O capítulo 5 faz um estudo detalhado sobre o escopo de pesquisa, envolvendo o projeto PERTMED, o *middleware* EXEHDA, o EXEHDA-SS que é uma recente contribuição ao Subsistema de Reconhecimento de Contexto e Adaptação o mesmo será parte importante a ser incorporado ao modelo proposto.

No capítulo 6 é tratada a concepção e a modelagem do uMED, são tratados aspectos da modelagem e concepção da arquitetura de software do uMED.

O capítulo 7 é tratado as tecnologias utilizadas e um estudo de caso do uMED, são apresentadas as principais tecnologias empregadas no uMED e o estudo de caso afim de explorar o suporte oferecido pelo uMED.

No capítulo 8 é discutido as considerações finais, são apresentadas as principais contribuições pertinentes deste trabalho, discussão dos trabalhos relacionados ao uMED, trabalhos futuros e publicações realizadas.

Por sua vez o anexo A faz um estudo sobre padrões de sensores e atuadores.

O anexo B com o intuito de caracterizar sinais vitais de pacientes, apresentas os principais sinais vitais e suas faixas médias.

2 COMPUTAÇÃO UBÍQUA

De acordo com Costa (COSTA; YAMIN; GEYER, 2008) a computação ubíqua idealizada por Mark Weiser na histórica publicação (WEISER, 1991) é atualmente um dos modelos de computação que mais dissemina na atualidade. Weiser postulava que a computação residiria nos mais inusitados objetos (e.g., etiquetas de roupas, xícaras de café, interruptores de luz e canetas) tornando-se, muitas vezes, imperceptível aos olhos dos usuários. Neste contexto, onde a computação torna-se imersa ao cotidiano, as pessoas convivem com os computadores e não somente os utilizam. Nestes ambientes, ditos ubíquos, existiria uma intensa interação entre os dispositivos que o compõem com a finalidade de auxiliar os usuários na execução de suas tarefas.

Na computação ubíqua os recursos se adaptam ao comportamento humano de modo não intrusivo, sem forçar que os usuários se adaptem aos dispositivos (GOULARTE, 2003). A proposta de Weiser vem se tornando realidade, através de tecnologias como PDAs, *Smartphones* e a consolidação de padrões para redes sem fio como o *Bluetooth* e o IEEE (*Institute of Electrical and Electronics Engineers*) 802.11. Com a computação ubíqua, a relação entre usuários e dispositivos computacionais muda em relação aos computadores pessoais, o que era de um para um, passa a ser de um para muitos (um usuário para vários dispositivos) (WEISER; GOLD; BROWN, 1999).

A computação ubíqua vem conquistando espaços com a disseminação de dispositivos portáteis, sobretudo após a ampliação das tecnologias de redes sem fio. Dessa forma, o uso desta computação tem conduzido a aplicações de diversas áreas do mundo real, onde sua usabilidade nos moldes da computação tradicional não poderia ser tão ampla. Cenários na área médica, podem estender-se a diversas atividades, como telemonitoramento através do uso de sensores de medicina ubíqua, diagnósticos remotos, segunda opinião, juntas médicas, *Home Care* viabilizando a integração de dispositivos móveis como PDAs, *Tablets, Smartphones* com os sistemas de prontuário eletrônico existentes nos hospitais, entre outros cenários, desde residências inteligentes a trabalhos colaborativos (DINIZ, 2009).

A premissa principal por trás do conceito de computação ubíqua é desenvolver uma variedade de dispositivos inteligentes para serem utilizados em ambientes de trabalho e/ou residencial. Tais dispositivos provêm aos usuários acesso universal e imediato às informações e dão suporte às tarefas dos usuários (GRIMM; BERSHAD, 2002).

A figura 2.1 apresenta um esquema para computação ubíqua. Além de mobilidade, os sistemas ubíquos requerem suporte à escalabilidade, interoperabilidade, sensibilidade dentre outros aspectos para garantir que os usuários tenham acesso quando necessitarem (SAHA; MUKHERJEE, 2003).

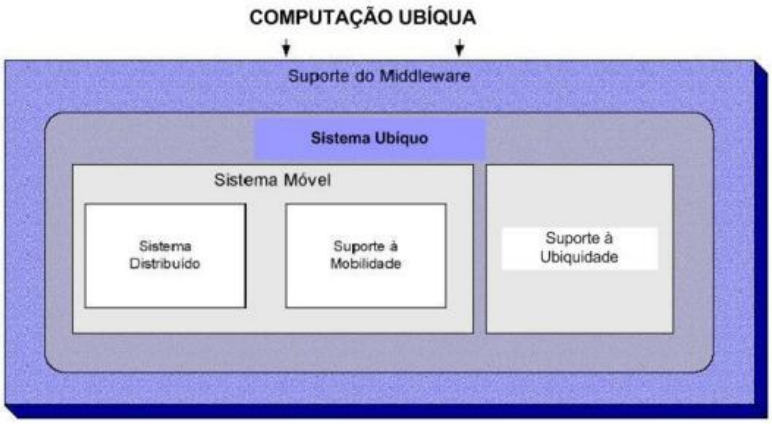

Figura 2.1: Computação Ubíqua (SAHA; MUKHERJEE, 2003)

Conforme e Mukherjee (SAHA; MUKHERJEE, 2003), os avanços tecnológicos necessários para construir um ambiente de computação ubíqua são os seguintes: rede de interconexão, dispositivos, *middleware* e aplicações (vide figura 2.2).

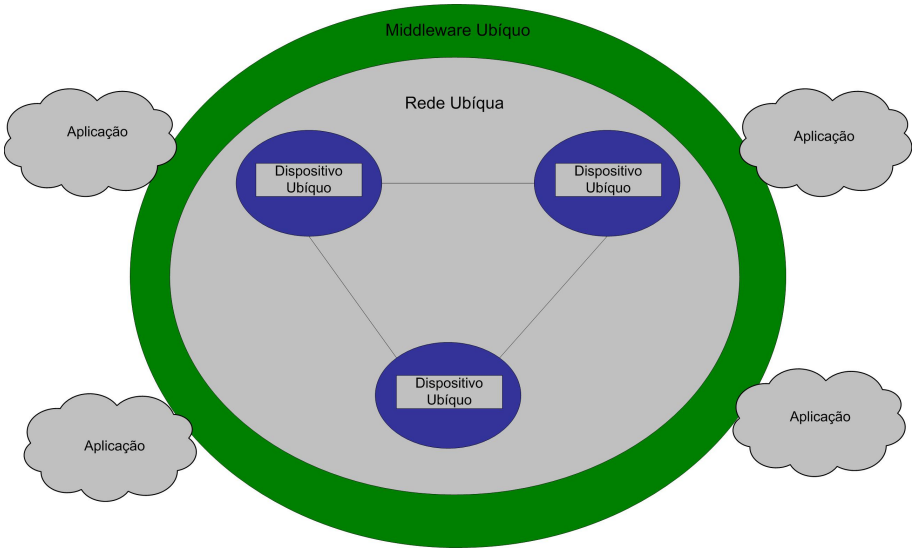

Figura 2.2: Framework para Computação Ubíqua (adaptada de (SAHA; MUKHERJEE, 2003))

Segundo Johnson (JOHNSON, 2005), os ambientes de computação ubíqua são espaços inteligentes, contendo dispositivos móveis sem fio, interconectados entre si, com consciência das informações do ambiente e reagindo de forma inteligente as informações do mesmo, disponível a qualquer hora e lugar. Para Couloris e colegas (COULORIS; DOLLIMORE; KINDBERG, 2005), um espaço inteligente é qualquer espaço físico com serviços embarcados, ou seja, serviços providos sobretudo dentro daquele espaço físico.

Já as aplicações ubíquas, em uma visão mais ampla, devem prever a mobilidade de equipamentos e usuários, denominada mobilidade física, e também dos componentes da aplicação e serviços, chamada de mobilidade lógica. Portanto, as aplicações devem ter o estilo siga-me, facultando que o usuário possa acessar seu ambiente computacional

independente da localização, do tempo e do dispositivo utilizado (YAMIN, 2004). Para tal, as aplicações precisam entender e se adaptar ao ambiente, compreendendo o contexto em que estão inseridas (MACIEL; ASSIS, 2004). Essa nova classe de sistemas computacionais, sensíveis ao contexto, abre perspectivas para o desenvolvimento de aplicações muito mais ricas, elaboradas e complexas, que exploram a natureza dinâmica e a mobilidade do usuário. Um desafio central na programação deste tipo de aplicação é possibilitar que as mesmas se adaptem continuamente ao ambiente e permaneçam funcionando mesmo quando o indivíduo se movimentar ou migrar para outro dispositivo (GRIMM; BERSHAD, 2002; COSTA; YAMIN; GEYER, 2008)

De acordo com Augustin (AUGUSTIN, 2004) a computação ubíqua, constitui um ambiente altamente distribuído, heterogêneo, dinâmico, móvel, mutável e com forte interação entre homem e máquina.

A computação ubíqua vem sendo estudada por diversos grupos de pesquisas em todo o mundo. No entanto, muitos desafios são identificados para que o paradigma da ubiquidade seja alcançado. Alguns destes desafios são de interesse direto deste trabalho, e serão apontados nos próximos capítulos.

2.1 Características da Computação Ubíqua

Para satisfazer os requisitos definidos pela computação ubíqua, uma série de características são necessárias para particularizar um sistema ou ambiente ubíquo. Este são objetivos fundamentais recorrentes em projetos de pesquisa efetuados na área, seja no meio acadêmico, ou no meio empresarial (em centros tecnológicos e laboratórios de empresas) (GREGORY; ELIZABETH, 2000; DAVID, 2007).

Características de uma ambiente ubíquo (YAU; YU; KARIM, 2002; COSTA; YAMIN; GEYER, 2008):

- onipresença dos serviços: permitir a movimentação física do usuário, dando a ele a percepção de estar levando consigo os serviços computacionais;

- invisibilidade: não deve ser interpretada literalmente. Seu significado neste contexto é de tornar a computação não perceptível no foco de atenção do usuário. O usuário deve concentrar-se na tarefa, não no computador. A invisibilidade não é uma questão apenas de interface de *software*, ela envolve a forma como visualizamos a tecnologia, como interagimos com ela e como a utilizamos para obter os resultados desejados;

- sensibilidade ao contexto: capacidade de coletar informações sobre o ambiente onde está sendo utilizado isto pode gerar informações relevantes a respeito do ambiente, o excesso desta, caso não corretamente filtrada e interpretada, pode ser prejudicial ao meio. O gerenciamento adequado das informações sobre o ambiente, assim como a disponibilização e correta utilização das mesmas são pontos fundamentais para o sucesso de qualquer ambiente ubíquo;

- captura de experiências: capacidade de capturar e registrar experiências para uso posterior, se um sistema pode identificar usuários, objetos e dispositivos inseridos em um meio e conhecer a sua localização, então o potencial para gerar informações

a respeito deste ambiente é elevado. Este potencial pode ser utilizado para adequar o meio a uma nova realidade ou estado, ou até mesmo para reconhecer padrões de comportamento de usuários e, de forma pró-ativa, se antecipar a comandos explícitos do mesmo;

- comportamento adaptável ou dinamismo de tarefas: capacidade de, dinamicamente, adaptar os serviços disponíveis ao ambiente onde está sendo utilizado dentro de suas limitações;

- descoberta de serviços: construir serviços pró-ativamente, de acordo com o ambiente em que se encontra. A aplicação deve interagir com o ambiente e permitir que o usuário também o faça, a fim de descobrir novos serviços ou informações para atingir o objetivo desejado;

- composição de funcionalidades: capacidade de, a partir de serviços básicos, montar uma determinada funcionalidade requerida pelo usuário;

- interoperabilidade espontânea: capacidade de alterar os "parceiros" durante a sua operação conforme a sua movimentação;

- heterogeneridade de dispositivos: prover mobilidade da aplicação através de dispositivos heterogêneos;

- tolerância a falhas: capacidadade de se adaptar diante de falhas no ambiente (por exemplo, disponibilidade *on-line/off-line*).

2.2 Cenários em Computação Ubíqua

Este seção resume o estudo desenvolvido a respeito da origem da computação ubíqua. A computação ubíqua é um modelo computacional proveniente das tecnologias de rede sem fio e sistemas distribuídos, em um processo evolutivo iniciado pela computação nômade e seguido pela computação móvel, estágio atual da tecnologia móvel (COSTA; YAMIN; GEYER, 2008). A ideia deste modelo é a criação de um ambiente físico onde o foco é o usuário, especificamente a tarefa que ele deseja realizar, facultando assim ao usuário dedicar-se às questões de maior importância, permitindo assim que o ambiente ubíquo encarregue-se da execução de tarefas secundárias (AUGUSTIN et al., 2008).

Pelo fato da computação ubíqua ser uma área emergente de pesquisa, termos como computação ubíqua, computação pervasiva, computação nômade, computação móvel e outros tantos têm sido usados muitas vezes como sinônimos, embora sejam diferentes conceitualmente e utilizam diferentes ideias de organização e gestão dos serviços computacionais. À medida que a área evolui, esses conceitos vão sendo melhor compreendidos e as suas definições tornam-se mais evidentes e copiosamente utilizadas (AUGUSTIN et al., 2008).

Os sistemas de computação móvel ainda não estão bem caracterizados e este termo é usado pelos autores em um espectro de ambientes, que envolvem alguma forma de mobilidade. De forma geral, pode-se dizer que sistema de computação móvel é um sistema distribuído que envolve elementos (*software*, dados, *hardware*, usuário) cuja localização

se altera ao longo da execução (AUGUSTIN et al., 2008). Esta definição torna clara a amplitude desta nova área de computação.

Dependendo dos elementos que possuem a propriedade de mobilidade, podem-se definir diferentes cenários. Entre eles destaca-se (AUGUSTIN et al., 2008):

- computação nômade - democratizada pelo uso de dispositivos portáteis tais como os *palmtops* e suas aplicações de gerenciamento pessoal. O usuário podia utilizar os serviços que um computador oferecia independentemente da sua localização. A mobilidade está mascarada através da portabilidade do *hardware*. No início dos anos 90, as facilidades de comunicação eram, basicamente, via acesso discado a cada movimentação, uma nova conexão à rede era necessária;

- computação via redes sem fio - usuário utilizando um equipamento portátil pode se deslocar dentro de uma célula, enquanto mantém a conexão à rede fixa ou à uma rede espontânea (*ad-hoc*) que se forma pelo encontro de dispositivos;

- mobilidade de código - os componentes da aplicação podem se mover. Pode-se ter: a mobilidade de código; a mobilidade de dados; ou a mobilidade de todo o estado da execução da aplicação (por exemplo: agentes móveis);

- computação móvel - a computação nômade, combinada com a capacidade de acesso permanente à rede sem fio, tem transformado a computação numa atividade que pode ser transportada para qualquer lugar. Observa-se que a crescente introdução de facilidades de comunicação tem deslocado as aplicações da computação móvel de uma perspectiva de uso pessoal para outras mais avançadas e de uso corporativo, como as aplicações móveis distribuídas;

- computação pervasiva - nesta concepção, o computador tem a capacidade de obter informação do ambiente no qual ele está inserido e utilizá-la para dinamicamente construir modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação para melhor atender as necessidades do dispositivo ou utilizador sendo assim, adaptação consciente do contexto. O ambiente também pode e deve ser capaz de detectar outros dispositivos que venham a fazer parte dele. Desta interação surge a capacidade dos sistemas agirem de forma inteligente no ambiente no qual o usuário se movimenta, no qual esse ambiente está repleto de sensores e serviços computacionais;

- computação ubíqua - o ambiente é formado por inúmeros dispositivos móveis ou fixos e equipamentos computacionais conectados entre si e invisíveis ao usuário final. O usuário dispõe de seu ambiente computacional independente de localização, tempo, dispositivo e rede subjacente. Surge dos avanços da computação móvel e da computação pervasiva, e da necessidade de integrá-las. Isto significa que, qualquer objeto computacional presente no ambiente ou trazido pelo usuário pode construir dinamicamente modelos computacionais dos ambientes entre os quais o usuário se movimenta e os serviços por ele utilizado são autoconfiguráveis dependendo da necessidade e da tarefa que o usuário deseja realizar.

Muitos pesquisadores consideram que computação pervasiva, termo cunhado pela IBM (2000), e computação ubíqua, proposto por Mark Weiser - XEROX Parc (*Ubiquitous*

Computing) (WEISER, 1991), como sinônimos (SATYANARAYANAN, 2001). Observando os trabalhos realizados por pesquisadores, a maioria destes usam os termos de forma indistinta e neste trabalho ocorre o mesmo.

A computação ubíqua pode ser definida como a integração entre a mobilidade, sistemas de reconhecimento de contexto e computação distribuída de forma invisível ao usuário.

2.3 Tendências em Computação Ubíqua

Segundo Augustin (AUGUSTIN et al., 2008) a computação ubíqua também chamada de tecnologia tranquila (*Calm Tecnhology*), Inteligência Ambiente (*Inteligence Ambient*), Computação Pró-ativa (*Proactive Computing*), *Internet* dos Objetos (*Internet of Things*) e Computação Invisível (*Invisible Computing*) entre outros nomes. Porém, os termos que têm prevalecido são computação pervasiva e computação ubíqua.

Em um ambiente ubíquo ou espaço ubíquo ou também chamado de *smart space*, computadores e outros dispositivos digitais estão totalmente integrados ao ambiente do usuário e objetivam auxiliá-lo em suas tarefas diárias. Este é um ambiente altamente dinâmico e heterogêneo. Os recursos, incluindo serviços, dispositivos e aplicações, disponíveis podem alterar-se rapidamente. Espaços heterogêneos têm diferentes tipos de recursos disponíveis e diferentes políticas de uso dos recursos. Programas em execução neste ambiente devem ser capazes de se adaptar à troca do contexto e disponibilidade de recursos. Isto coloca um desafio para os desenvolvedores que devem especificar como o programa deve se comportar em distintos contextos e quando diferentes tipos de recursos estão disponíveis. Além disso, ambientes ubíquos podem ter diferentes modos de executar a mesma tarefa devido a possibilidade de alterar os serviços ou a aplicações ou até mesmo os recursos. O desenvolvedor não pode prever como as várias tarefas serão executadas nos diferentes espaço ubíquos. Assim, programadores necessitam de abstrações de alto nível para programar aplicações em um ambiente ubíquo sem precisar ter consciência dos recurso disponíveis, contexto, políticas e preferência dos usuários (AUGUSTIN, 2004).

Devido a crescente evolução em sistemas de comunicação e computação, está se tornando frequente a integração da computação as atividades humanas. Microprocessadores estão cada vez menores e baratos o suficiente para serem colocados em quase tudo - não somente em dispositivos digitais, carros, eletroeletrônicos, brinquedos, ferramentas, mas também em objetos (lápis, por exemplo) e roupas. Todos esses artefatos devem estar interligados e conectados em uma rede sem fio (AUGUSTIN et al., 2008).

De fato, a tecnologia espera uma revolução na qual bilhões de pequenos e móveis processadores estejam incorporados ao mundo físico, compondo objetos "espertos" sabem onde estão, se adaptam ao ambiente e fornecem serviços úteis em adição ao seu propósito original, formam redes espontâneas (*ad-hoc*) e altamente distribuídas, numa escala exponencial sem precedentes (AUGUSTIN et al., 2008).

Esse cenário está sendo considerado como um paradigma inovador do século XXI (SAHA; MUKHERJEE, 2003; SATYANARAYANAN, 2001), ou a terceira onda da computação (JANSEN, 2005), o qual permite o acoplamento do mundo físico ao mundo da informação e fornece uma abundância de serviços e aplicações onipresentes visando que usuários, máquinas, dados, aplicações e objetos do espaço físico interajam uns com os outros de forma transparente (em *background*) (RANGANATHAN, 2005).

Pela integração de sensores, computadores, dispositivos e redes foi possível dese-

nhar a primeira geração de ambientes ubíquos, referenciados como ambientes integrados. Agora os esforços de pesquisa concentram-se em deslocar o paradigma de ambientes integrados para espaços programáveis. O grande desafio é que a computação ubíqua afeta toda a área da ciência da computação em três distintas perspectivas: da experiência, da engenharia e teórica (CHALMERS, 2006).

Modelos de programação e *middlewares* baseados no modelo de sensores-atuadores-contexto foram os mais focados para a programação de aplicações da primeira geração e resultaram no conceito de computação orientada a contexto (*context-aware-programming*). Busca-se, agora, encontrar abstrações de alto nível que permitam programar aplicações que farão parte do paradigma denominado programação orientada à tarefas (*Task-Oriented Programming*) (AUGUSTIN et al., 2008).

2.4 Considerações sobre o Capítulo

Este capítulo apresentou uma breve introdução à computação ubíqua, bem como os requisitos que caracterizam um ambiente ubíquo. Assim como, a origem da computação ubíqua e os seus principais aspectos de estudo e pesquisa, também foi apresentado uma perspectiva sobre a evolução da computação ubíqua. No próximo capítulo é apresentado à medicina ubíqua identificando características e tecnologias.

3 MEDICINA UBÍQUA: IDENTIFICANDO CARACTERÍSTICAS E TECNOLOGIAS

A medicina ubíqua se potencializa com o aumento do uso das facilidades tecnológicas, dentre as quais destacam-se os dispositivos móveis e as redes de comunicação sem fio, as quais trazem novas possibilidades de acesso e interação para os seus usuários, como já discutido anteriormente.

O acesso as informações dos pacientes como, por exemplo, resultados de exames laboratoriais podem ser mais relevante quando o médico estiver perto do paciente que quando ele estiver em qualquer outro lugar (DINIZ, 2009).

Rodriguez e colegas (RODRIGUEZ, 2004) descrevem um sistema de informações médicas desenvolvido para permitir acesso a registros de pacientes baseado na localização do usuário. O sistema baseia-se em dispositivos *handhelds* usando estimativas de localização do usuário que acessa as informações do sistema hospitalar que sejam relevantes para a sua localização, por exemplo, quando o médico estiver próximo ao paciente ele terá acesso aos exames daquele paciente, entretanto se ele não tiver próximo ao mesmo, poderá não ter tal acesso.

O trabalho desenvolvido pelo grupo de Rodriguez trata de uma aplicação com caráter específico, mas não uma solução genérica, baseada em *middleware*.

Segundo Struzik (STRUZIK, 2007) o diagnóstico médico e tratamentos são geralmente precedidas por uma avaliação de sinais e sintomas do paciente. Os sinais, e as mudanças de variáveis físicas ou estados do paciente, são informações objetivas utilizadas pelos médicos, por exemplo, arritmia, taquicardia e febre. Os sintomas, por outro lado, são na sua maioria subjetivos e relatados pelos pacientes, por exemplo, tonturas, cansaço e dor. Tradicionalmente, informações objetivas, bem como informações subjetivas são obtidas em instalações específicas, tais como: clínicas médicas, hospitais. Todavia tecnologias emergentes, como computadores portáteis, redes de alta velocidade e biossensores médicos, estão mudando este ambiente dito tradicionais. Agora o sinal do pacientes pode ser captado remotamente. A disponibilidade de computadores portáteis e programas específicos permite ao médicos avaliar os sintomas psíquicos e somáticos no momento em que estes ocorrem. Desta forma, as informações médicas necessárias para o cuidado e tratamento do paciente estão se tornando onipresente.

3.1 O Ambiente de Medicina Ubíqua

Ubiquitous healthcare é um campo emergente da Tecnologia de Informação que utiliza um grande número de sensores e atuadores para monitorar o paciente capaz de melhorar a sua condição física e mental (BROWN; ADAMS, 2007).

Conforme (BROWN; ADAMS, 2007) pequenos sensores estão sendo projetados para recolher informações sobre as condições corporais, como temperatura, frequência cardíaca, pressão arterial, níveis químicos do sangue e da urina, frequência respiratória e níveis de atividade. Estas informações podem ser utilizadas para diagnosticar problemas de saúde. Os sensores usados, são implantados no corpo, ou instalados em suas residências e locais de trabalho.

Os atuadores poderão ir mais longe e desencadear ações como a liberação de pequenas quantidades de produtos farmacêuticos para a corrente sanguínea ou a estimulação elétrica de áreas do cérebro (por exemplo, aqueles implicados em condições tais como a doença de Alzheimer e de Parkinson ou aqueles associados com depressão).

O principal objetivo destes sensores e atuadores é ajudar os pacientes e as pessoas que a cuidam a monitorar o estado de saúde, auxiliando na elaboração de cuidados médicos e implementação de intervenções para melhorar a situação. Inicialmente, eles tendem a ser utilizados por médicos de família para controlar remotamente os pacientes, e fornecer conselhos de saúde em geral. Isto é particularmente útil para pacientes com mobilidade prejudicada, tipicamente idosos. Com o tempo, a tecnologia destina-se a apoiar um maior autocontrole e cuidado por todos os indivíduos, e não apenas aqueles em condições crônicas. Pacientes como crianças e aqueles com deficiências cognitivas, será necessário um apoio mais intensivo de profissionais da saúde e familiares. Nós últimos anos uma área denominada *ubiquitous healthcare technologies* se propõe monitorar e aconselhar sobre fatores de saúde a longo prazo, como dieta e exercício, aconselhando a uma mudança no sentido de "bem-estar" que incorpora, bem-estar como saúde física e mental (BROWN; ADAMS, 2007).

As tecnologias de computação ubíqua estão sendo usadas para melhorar o desempenho do paciente, dispositivos de apoio - como cadeira de rodas inteligentes que evitam o impacto com objetos e, especialmente, com outras pessoas em áreas congestionadas, e fornecem um *feedback*, como descrições verbais de objetos para deficientes visuais.

Conforme (BROWN; ADAMS, 2007) tecnologias também estão sendo desenvolvidas para apoiar as atividades dos profissionais da área de saúde. Exemplos incluem sistemas de prontuário do paciente que modificam as informações apresentadas com base no seu contexto atual provendo um apoio para melhorar o fluxo de informações entre os profissionais de saúde durante as mudanças de turno. Outro exemplo, seria a transmissão de informações (incluindo imagens) para o hospital de uma possível vítima de um acidente no intervalo de tempo até a chegada da ambulância no hospital. Sistemas foram também desenvolvidos para apoiar a formação de médicos.

Ambientes de medicina ubíqua de grande porte, como hospitais, o conceito de espaços inteligentes deverá possibilitar a realização das seguintes ações (DINIZ, 2009):

- conceder aos profissionais de saúde, acesso às informações do PEP, de qualquer local, desde que estejam conectados à rede e utilizando dispositivos móveis ou fixo. O ambiente se encarregará de gerenciar a comunicação de usuários e dispositivos às fontes de informações;

- permitir que uma aplicações possa ser transferida a outro dispositivo, dando continuidade a sessão;

- contolar as sessões de PEPs de diferentes usuários, usando diferentes dispositivos e redes de acesso;

- empregar informações de contexto para cooperar na disseminação da informação, automatização de configuração e adaptação de conteúdo.

3.1.1 Hospital do Futuro: Idéias e Conceitos

O hospital do futuro é uma visão de um hospital altamente interativo, onde os médicos podem acessar informações médica relevantes e colaborar tanto com os colegas, como com seus pacientes independentemente de tempo e lugar. Outros pesquisadores têm utilizado o termo "hospitais inteligente" (MITCHELL, 2000), outros, no entanto, preferem o termo "hospital interativo" (BARDRAM, 2003), já o projeto PERTMED usa o termo "hospital virtual" (PERTMED, 2007), neste trabalho todos os termos são considerados sinônimos e portanto são utilizados indistintamente.

Acredita-se que, a fim de criar um ambiente hospitalar verdadeiramente interativo precisa-se conceber um novo tipo de infraestrutura básica sobre a qual os sistemas clínicos informatizados podem ser projetados, desenvolvidos e implantados. Como parte do núcleo do ambiente de execução, esta infraestrutura deve suportar os aspectos do trabalho médico já introduzidos na seção 3.1, e deve funcionar como uma linha mestra computacional de um hospital. A infraestrutura deve suportar que os clínicos possam se movimentar livremente dentro e fora do hospital, mantendo o seu ambiente computacional intacto. Os clínicos devem ser capazes de se mover quando iniciar, pausar, reiniciar e suspender a sua interação com os sistemas clínicos informatizados. A infraestrutura também deve assegurar a adequada interoperabilidade dos diferentes sistemas em uso clínico de um hospital, fornecendo mecanismos básicos para desenvolvedores de sistemas clínicos para criar sistemas altamente integrados (BARDRAM, 2003).

Os temas relevantes para infraestrutura pretendida conforme Bardram e colegas (BARDRAM, 2003; BARDRAM; BOSSEN, 2005):

- adaptação - a infraestrutura deve apoiar o trabalho dos clínicos e tentar abranger e se adaptar a diversos serviços necessários em um ambiente heterogêneo. Por exemplo, um radiologista chegar a uma sala de conferência deve ser capaz de transferir toda a sua conferência de radiologia do seu PDA a um equipamento que ofereça uma tela de grandes dimensões;

- segurança - essa infraestrutura obviamente precisa incorporar mecanismos de segurança, porque trata com informações confidencias de pacientes entre outras informações sigilosas;

- sensibilidade ao contexto - acredita-se que a criação de sistemas clínicos sensíveis ao contexto é fundamental para sua aprovação no trabalho clínico. Em um ambiente de trabalho frenético, onde muitas tarefas são realizadas em paralelo, onde os médicos se deslocam de um lugar para outro, e onde as configurações de colaboração emergem constantemente, é importante que aplicações clínicas tenham um conhecimento sobre o contexto de trabalho do usuário e que eles sejam capazes

de se adaptar a este contexto. Apoio à sensibilidade ao contexto deve ser parte da infraestrutura básica, ao invés de algo que cada aplicação clínica implemente;

- colaboração - pretende-se criar mecanismos para a colaboração na infraestrutura, porque a colaboração é fundamental para a clínica de trabalho. Isso pode ser utilizado para a criação de conferências entre médicos e pacientes em casa ou entre médicos, facilitando a obtenção de uma segunda opinião.

3.1.2 Dispositivos Interativos em Ambientes Hospitalares

Segundo Bardram e colegas (BARDRAM; BOSSEN, 2005; BARDRAM; CHRISTENSEN, 2007) computadores pessoais são feitos para trabalho de escritório e são, portanto, muitas vezes difíceis de implantar e usar em um hospital, onde não há mesas apropriadas para um computador e dispositivos de informática. Portanto, as aplicações clínicas informatizadas devem ser incorporadas ao equipamento médico e hospitalar como um todo. Há trabalhos em protótipos para criar paredes, tetos e pisos iterativos, bem como a incorporação de recursos computacionais em leitos hospitalares, nos recipientes de comprimidos, nos instrumentos cirúrgicos, entre outros (vide figura 3.1).

Bardram (BARDRAM, 2003) prevê um hospital onde os médicos podem se aproximar de superfícies interativas em qualquer lugar e continuar seu trabalho. Algumas das superfícies são pequenas e portáteis, como PDAs (mas não são pessoais), outros são grandes, como as utilizadas em uma sala de conferências de radiologia, onde uma parede inteira pode ser uma grande superfície interativa.

Os registros médicos são em grande parte textual. Isto é uma forma abstrata de representar o saber médico. Coisas simples como usar o vídeo para documentar a reabilitação ortopédica, ou a cicatrização de feridas é desejável, mas não é facilmente integrado nos sistemas existentes. Outra parte importante desta pesquisa realizada por Bardram e sua equipe é desenvolver novas formas de interação com dados médicos e novos tipos de equipamento, o mouse é difícil de usar na cabeceira, por exemplo. Deste modo prevê-se o uso de vários tipos de interação multimodal. Por exemplo, permitindo ao cirurgião o acesso a prontuários e imagens de Raio-x usando voz e gestos, durante a execução da operação (BARDRAM, 2003; BARDRAM; BOSSEN, 2005).

Figura 3.1: Dispositivos Interativos (BARDRAM; CHRISTENSEN, 2007)

3.1.3 Contribuições para Área da Saúde

Conforme visto anteriormente o uso da computação ubíqua aplicado a medicina pode oferecer vantagens tais como: aumento na qualidade e eficiência dos serviços e melhora no gerenciamento da relação com o paciente (VARSHNEY, 2003). Este novo

sistema de saúde também prevê uma visão de "hospital virtual", o qual estende-se para casa dos pacientes ou lugares onde eles se encontram, onde sensores/dispositivos monitoram as condições ambientais e do paciente e comunicam-se, via rede sem fio, com as centrais médicas para a tomada de decisões e ações pertinentes. Experiências nesse sentido estão sendo conduzidas por alguns projetos de pesquisa europeus, como o do *Center of Pervasive Healthcare* da Dinamarca que desenvolve o projeto *Hospital of the Future* (BARDRAM; BOSSEN, 2005).

Estes indicadores prevêem que a computação ubíqua terá um enorme potencial de aplicabilidade na área da saúde. Esta situação é reforçada por projetos bastante atuais. Segundo o projeto PERTMED alguns dos problemas na área de saúde hoje em dia são (PERTMED, 2007):

- dificuldade de acesso a serviços especializados em regiões remotas ou carentes;
- custo elevado para o transporte de pacientes, especialmente de áreas pobres e rurais;
- interrupções e atrasos na sequência do tratamento.

Segundo o projeto PERTMED a questão que permeia esses três problemas é o acesso a informação de onde ela é gerada para onde ela é necessária, em tempo razoável e compatível com a gravidade da situação sendo tratada. A rapidez da decisão médica depende da pronta disponibilidade destas informações, sendo este o fator decisivo para a qualidade dos serviços prestados. Acesso à informação pode ser usado para substituir o transporte, por exemplo, um "paciente virtual" (formado por um conjunto de informações sobre seu estado de saúde) pode ser acessado a longas distâncias por um especialista remoto (PERTMED, 2007).

Com a introdução da computação ubíqua na área da saúde, dentre outros aspectos, é possível identificar uma contribuição no sentido de superação de desigualdades regionais e socioeconômicas, relativas ao acesso às informações dos sistemas de saúde.

3.2 Telemedicina e Medicina Ubíqua: Revisando Conceitos

Cabe ressaltar que medicina ubíqua e telemedicina contemplam aspectos que se sobrepõem. Inclusive alguns trabalhos os utilizam de forma indistinta. Porém, algumas particularidades devem ser consideradas.

Conceituar telemedicina, tem se tornado cada vez mais pauta de discussões acadêmicas, não tanto pelo conceito em si, mas, e principalmente, pela tentativa de classificação e aplicação das diferentes iniciativas existentes no mercado. Várias definições foram propostas por diferentes entidades e autores, com pontos de convergência, mas, muitas vezes, com amplitudes diferentes (VIEGAS, 1998; WOOTTON; CRAIG; PATTERSON, 2006).

Algumas das principais definições segundo Medeiros (MEDEIROS, 2009) são:

- Organização Mundial de Saúde - segundo a OMS, Telemedicina "é a oferta de serviços ligados aos cuidados com a saúde, nos casos em que a distância é um fator crítico. Tais serviços são providos por profissionais da área da saúde, usando tecnologias de informação e de comunicação para o intercâmbio de informações válidas

para diagnósticos, prevenção e tratamento de doenças e a contínua educação de provedores de cuidados com a saúde, assim como para fins de pesquisas e avaliações. O objetivo primeiro é melhorar a saúde das pessoas e de suas comunidades";

- *American Telemedicine Association* - segundo a ATA, Telemedicina "é a utilização de informação médica transmitida de um local a outro através de meios de comunicação eletrônica, visando o cuidado com a saúde e a educação do paciente ou do provedor de cuidados com a saúde, com o propósito de melhorar o cuidado com o paciente";

- *National Air and Space Agency* - segundo a NASA, Telemedicina "é a integração de tecnologias de telecomunicações, de informação, de interface homem-máquina e de cuidados médicos com o propósito de melhorar os cuidados com a saúde em vôos espaciais".

Estes conceitos são resultados do reflexo direto do uso do conceito por médicos, pesquisadores clínicos, entre outros profissionais, que a partir do avanço das tecnologias da informação e comunicação, passaram a apoiar os cuidados de saúde. Para exemplificar o que é telemedicina, tem sido comum as interpretações mais simples, como o uso do telefone pelos pacientes na consulta os seus médicos, ou o processo de rádiocomunicação entre uma unidade móvel como uma ambulância e a emergência do hospital, em que a primeira informa sinais e sintomas dos pacientes de forma que, do outro lado, o médico inicie um processo diagnóstico e a recomendação de tratamento (WOOTTON; CRAIG; PATTERSON, 2006).

Segundo o Laboratório Nacional de Computação Científica (LNCC, 2008) a telemedicina teve suas origens no início do século XX. Entretanto, isto está atualmente ultrapassado frente às possibilidades que surgem com a utilização das novas tecnologias. Tais recursos tecnológicos, ainda não estão totalmente disponíveis nos sistemas de saúde, seja ele, público ou privado. A sua implantação depende, de uma maneira geral, da integração através de uma infraestrutura computacional inserida em um "ciberambiente" (grid de computação distribuída de alto desempenho) onde equipes multidisciplinares altamente competentes em diversas áreas do conhecimento humano promovam mecanismos de transferência e inovação na solução deste importante problema nacional que é a saúde.

Com a aplicação das características da ubiquidade (vide capítulo 2) na área da medicina, torna-se possíveis inúmeras aplicações que até então com o uso da telemedicina não eram exploradas dentre elas destaca-se: sensibilidade ao contexto, serviço de adaptação, descoberta de recurso entre outras (VENECIAN, 2010).

3.3 Medicina Ubíqua: Principais Aspectos de Estudo e Pesquisa

Nesta seção pretende-se caracterizar os principais desafios e algumas propostas de soluções que alguns autores encontram no uso da computação ubíqua na perspectiva da medicina ubíqua. A análise dos mesmos está dividida em três exemplos.

3.3.1 Caso 1 - Suporte à Mobilidade

Segundo Bardram e Bossen (BARDRAM; BOSSEN, 2005) para que um ambiente de computação ubíqua possa ser empregado eficientemente em tarefas clínicas, é fundamental que exista suporte a mobilidade e o trabalho colaborativo dos profissionais de saúde envolvidos. Em (BARDRAM; BOSSEN, 2005) são apresentadas estratégias vigentes para a realização desse suporte, sendo que basicamente tecnologias móveis são associadas a tecnologias de redes sem fio. Contudo, restam ainda muitos desafios a serem superados, para que o o uso desta tecnologia forneça subsídios de forma efetiva e clara ao trabalho clínico. A seguir os desafios que Bardram e Bossen destacam:

- **sobrecarga de informação** - devido ao fato dos profissionais de saúde tratarem de centenas de pacientes, cada um podendo dispor de dados clínicos volumosos, dificulta para os profissionais, que usualmente trabalham num ritmo acelerado, navegar por toda essa informação usando as pequenas telas dos dispositivos móveis;

- **heterogeneidade das tecnologias** - dispositivos móveis geralmente obriga os profissionais de saúde, que se movem constantemente trocando de dispositivo computacional em função de suas tarefas e localizações, a dificuldade e restabelecer suas sessões de usuário a cada troca de dispositivo;

- **isolamentos de dispositivos** - dispositivos móveis utilizados nos testes são intrinsecamente isolados e tendem a não suportar os frequentes trabalhos cooperativos *ad hoc* dos profissionais, já que não permitem a troca e o compartilhamento de visões e informações clínicas.

Segundo Bardram (BARDRAM, 2004) uma forma de amenizar a sobrecarga de informação é a ciência de contexto (*context-aware*), já que sistemas que usam essa tecnologia permite localizar e apresentar dados relevantes aos seus usuários, baseando-se em informações contextuais (e.g., identidade do usuário, o papel que este desempenha, a sua localização, o dispositivo que está usando, tamanho de tela do dispositivo). Com base na combinação dessas informações, aplicações podem adaptar-se a diferentes situações e comportar-se de forma perfeita para cada uma delas. Por exemplo, num hospital um médico carregando um PDA, ligado a um sistema ciente de contexto, ao aproximar-se do leito de um de seus pacientes, poderia obter os dados do PEP mostrados automaticamente neste PDA.

Conforme Bardram (BARDRAM, 2003) mobilidade de código é uma maneira de lidar com a heterogeneidade de dispositivos, já que suporta a transferência de sessão de usuário entre dispositivos (e.g., de *Desktop* para PDA), mas exige que a sessão seja adaptada durante essa transferência (e.g., o tamanho reduzido da tela do PDA e a largura de banda limitada da rede de acesso sem fio devem ser considerados). É importante ressaltar que num ambiente ubíquo os dispositivos móveis não devem estar isolados, mas sim inseridos à infraestrutura tecnológica existente.

Composição de dispositivos é um meio de lidar com o isolamento dos dispositivos móveis, já que permite a constituição *ad hoc* de um dispositivo lógico a partir de dispositivos físicos diferentes (PHAM et al., 2001). Por exemplo, um médico e um enfermeiro carregando seus respectivos *Tablets* poderiam colocá-los próximos, para que estes se compusessem num único *Tablet* lógico. Esses profissionais poderiam então baixar e

arrastar arquivos, objetos e aplicações entre as duas telas e interagir com os mesmos diretamente. Uma técnica correlata é o suporte à transferência de informações entre dispositivos heterogêneos (FAVELA et al., 2004), que possibilita, por exemplo, a transferência de informação médica de um computador com múltiplos usuários para um PDA pessoal, ou o uso desse PDA como ponto de acesso remoto a uma tela de grandes proporções na parede, ou ainda o uso de vários PDAs conectados a essa tela para suportar um conferência médica (BARDRAM, 2004).

3.3.2 Caso 2 - Suporte à Heterogeneidade

Em um cenário hospitalar por exemplo, onde médicos precisam acessar PEPs, de qualquer lugar, com uma grande diversidade de categorias de dispositivos móveis e fixos, ou seja, celulares, PDAs, *Tablets, Smartphones, Notebooks e Desktops* com possibilidade de transferência de sessão entre dispositivos com diferentes configurações, em tempo satisfatório e com consistência de informações. Para que esse ambiente ubíquo atendas a essa demandas é necessário superar alguns desafios conforme Diniz (DINIZ, 2009) destaca:

- **ambiente multiplataforma** - dispositivos com diferentes tamanhos de tela, sistemas operacionais, poder computacional entre outras características distintas;

- **autonomia limitada de alguns dispositivos móveis** - embora exista uma evolução das baterias utilizadas nos dispositivos móveis elas ainda possuem uma autonomia limitada. Muitas vezes, o médico começa a usar o seu dispositivo móvel quando o mesmo já está apenas com parte da sua carga total. Isto leva muitas vezes à necessidade de migração de sessão, durante a sua execução, devido à impossibilidade do médico conclui-la, já que o dispositivo utilizado não dispõe de energia suficiente para tal. Não é foco de estudo o problema de autonomia de baterias propriamente dito, porém pretende-se encontrar uma solução de *software* que permita que o médico ou outro profissional de saúde transfira sua sessão para outro dispositivo móvel ou fixo, imediatamente antes do término da bateria ou até mesmo após o término da mesma, sem perda de tempo nem conteúdo;

- **interrupção de uma sessão** - ruptura de acesso a um PEP, no momento em que a transferência de sessão entre dois dispositivos esteja ocorrendo. A transferência de sessão poderá acontecer por uma necessidade, como por exemplo, o aviso que a bateria está com a carga baixa em um dispositivo portátil ou por uma preferência de usuário. Em algumas situações, por exemplo, no ambiente hospitalar, num dispositivo com interface limitada, como um PDA ou *Smartphone*, é difícil de se visualizar o resultado de um exame de Raio-X de modo satisfatório. Nestes casos pode ser necessária a migração da sessão para um outro computador do tipo *Notebook* ou *Desktop*;

- **intermitência de comunicação** - em um ambiente ubíquo é possível existir diversas possibilidades de tecnologias de redes envolvidas no cenário, a exemplo de WiFi, *bluetooth*, IEEE 802.3, GSM, dentre outras. Essas diferenciações de tecnologias são ainda mais notáveis quando se trata de um ambiente de rede metropolitana, que ultrapassa os limites de um hospital ou uma clínica. O médico pode acessar o sistema de diferentes locais das cidade, como, de um hospital, de sua residência ou,

ainda, em trânsito. Sendo assim, poderá utilizar um acesso através de uma rede local, usando a *Internet* ou até mesmo a rede pública de telefonia celular. Neste caso o *handoff* (troca) entre diferentes tecnologias de comunicação é possível, conforme destacado no trabalho de Cavalcanti e colegas (CAVALCANTI, 2005). Essa variedade de enlaces de rede possibilita a interrupção de sessões de forma involuntária.

Segundo Diniz (DINIZ, 2009) a tarefa de transferência de sessão entre dispositivos é uma tarefa complicada e deve ser realizada, em muitos casos com o menor atraso possível, quando for requerida.

O médico terá que dar continuidade a uma sessão aberta em outro dispositivo, muitas vezes, em meio a uma atividade que depende de seu parecer. É necessário que a migração de sessões se dê em um intervalo que seja uma fração mínima daquele levado na troca de atividades pelo médico. A interrupção da migração também é um fator que deve ser considerado um possível desafio, porque a mesma poderá ser solicitada e não ser concluída devido à intermitência da rede ou escassez de bateria. A abordagem de migração de aplicações também vem sendo utilizada na área médica e ambientes ubíquos em geral. Como mencionado anteriormente, a migração de aplicação poderá ocasionar problemas quando for interrompida de forma involuntária. Segundo o autor não foram encontrados trabalhos relacionados que tratem dessa interrupção de migração de sessão, nem do tempo em que ela ocorre.

No ambiente hospitalar ubíquo, o tratamento a ser dado a estes problema pode ser obtido através de tarefas de gerenciamento de contexto e de sessão. Ações referentes ao gerenciamento de sessão permitem o armazenamento temporário da sessão e o repasse para outro dispositivo de destino. Para que haja essa migração sem perda de informação e de tempo, o contexto do dispositivo destino tem que ser conhecido previamente e controlado através de ações realizadas pelo serviço de gerenciamento de contexto. A abordagem usada para o tratamento da multiplataforma, notadamente no que diz respeito a variações de tamanho de tela, é a adaptação de conteúdo.

Conforme Diniz (DINIZ, 2009) para superar os desafios mencionados, é necessário definir uma arquitetura sensível ao contexto, para gerenciamento de sessões, possibilitando a migração de sessão e o seu resgate em outro dispositivo, em tempo hábil, e com adaptação de conteúdo, o tempo de troca de atividades será pequeno o suficiente para possibilitar um potencial aumento da produtividade do profissional cujas tarefas envolvam mobilidade e fragmentação, a exemplo do que ocorre em um ambiente hospitalar como mencionado anteriormente.

3.3.3 Caso 3 - Suporte à Ética Médica

Conforme Brown e colega (BROWN; ADAMS, 2007) a medicina ubíqua é uma área emergente da tecnologia que utiliza um grande número de sensores e atuadores, afim de monitorar e melhorar a condição física e mental dos pacientes. Pequenos sensores recolhem dados sobre quase todas as característica fisiológica que pode ser usado para diagnosticar problemas de saúde. Esta tecnologia enfrenta algumas questões éticas desafiadoras, que vão desde as questões de confiança e eficácia para as questões sociais da saúde e de lacunas relacionadas à situação econômica. Ela apresenta problemas particulares, combinando o desenvolvimento do computador - informação - comunicação social - ética médica. Isto levanta questões de saúde à medida que desenvolvem sistemas de saúde onipresente. Medicina é uma profissão cuja prática é restrita e controlada geralmente por

autoridades governamentais, nomeados, enquanto o *software* e desenvolvimento de *hardware* é notoriamente carente de tais regimes.

Além dos desafios tecnológicos envolvidos na medicina ubíqua existem ainda os desafios éticos no qual fogem ao escopo deste trabalho.

3.4 Considerações sobre o Capítulo

Este capítulo apresentou uma visão geral sobre medicina ubíqua caracterizando o seu ambiente assim como, as contribuições que podem serem providas para um ambiente de saúde, o hospital do futuro com suas peculiaridades, os dispositivos interativos empregados em ambientes médicos ubíquos, uma comparação entre telemedicina e medicina ubíqua, apresentou-se também alguns desafios desta área e maneiras de contorná-los. No próximo capítulo é apresentado alguns projetos de medicina ubíqua.

4 PROJETOS DE MEDICINA UBÍQUA: SISTEMATIZANDO SUAS CARACTERÍSTICAS

No levantamento do estado-da-arte, foram considerados projetos representativos da área de medicina ubíqua, os principais encontram-se resumidos neste capítulo. Os mesmos tem como elemento comum na implementação de suas propostas o emprego de um *middleware*. Genericamente estes *middlewares* têm o objetivo de prover abstrações para o desenvolvimento de aplicações em ambientes distribuídos e ubíquos atuando como camada intermediária entre o ambiente de execução e as aplicações.

4.1 Projeto ABC

O projeto ABC (*Activity-Based Computing*) desenvolvido na Universidade de Aarhus, na Dinamarca e conta com a participação da equipe do Hospital de Aarhus e demais pesquisadores (BARDRAM, 2003) (BARDRAM, 2004), (BARDRAM; CHRISTENSEN, 2007), (BARDRAM, 2009).

Segundo Voida (VOIDA, 2002) os princípios do projeto ABC são proveniente de pesquisas iniciadas em 2001 e tomam como base o conceito de *Activity-Based Computing* (ponto de partida da sigla ABC). Este conceito é um paradigma de interação e projeto que explora como os sistemas computacionais podem dar suporte direto a atividades específicas. Através da organização dos recursos em termos de atividades, as aplicações poderão, então manipulá-los, selecionando o mais importante para a sua tarefa .

O objetivo da computação baseado em atividades (ABC) (BARDRAM; CHRISTENSEN, 2007) é investigar a criação de suporte computacional para atividades humanas. Em contraste com as abordagens existentes, que tendem a focalizar os profissionais da informação sentado em uma mesa de escritório, a abordagem dos autores visa fornecer apoio computacional a atividade de trabalho clínico nos hospitais. Em um hospital, os desafios decorrentes da gestão de atividades paralelas e interrupções são potencializados e estão combinado com um elevado grau de mobilidade, colaboração e urgência.

No processo de desenvolvimento do projeto ABC tem como meta, atender cinco temas que refletem algumas ações desempenhadas diariamente em grandes centros hospitalares. São eles: controle e administração de medicamentos por enfermeiros, prescrição de medicamentos por médicos, colaboração, conferências e cirurgia (BARDRAM; CHRISTENSEN, 2007).

De acordo com Bardram e Christensen (BARDRAM; CHRISTENSEN, 2007) os cenários do dia-a-dia dos profissionais da área de saúde, verificando como as tecnologias

de *middleware* podem fornecer uma forte fundamentação para soluções ubíquas e móveis. Dentre os cenários indicados pelo grupo de pesquisa destacam-se os seguintes: prescrição médica, através de discussão entre profissionais de saúde sobre medicamentos e dosagens no tratamento de determinados pacientes, baseando-se no diagnóstico de exames; conversas explicativas sobre o diagnóstico e o tratamento com o próprio paciente através do uso de PDAs e aparelho de TV; busca por medicamentos na farmácia do hospital; videoconferência com teleconsulta para telediagnóstico.

O núcleo do *middleware* ABC, conforme figura 4.1, tem duas camadas de base na arquitetura. Uma camada de infraestrutura fornece suporte para persistência propagação do evento, e métodos para sincronizar atividades. Esta camada pode residir em um ou mais servidores e é independente da camada cliente. A camada de infraestrutura é responsável pela persistência de atividades através do componente *Activity Store*, para gerenciar estados de tempo de execução e eventos através do *Activity Manager* e para a distribuição de mudanças de estado e outros dados relacionados através *Collaboration Manager*. O componente *Context Service* mantém registro de usuários e máquinas, bem como seu contexto. A camada é normalmente empregado em um ou mais servidores fisicamente separada da camada de cliente (PEDERSEN; MOGENSEN; BARDRAM, 2006).

A camada cliente é executado em todos os dispositivos ativos e interfaces com o sistema operacional local, aplicações e dados. O *Activity Controller* fornece modos para retomar e suspender uma atividade, e gerenciar as conexões à infraestrutura bem como a outros clientes. É também dentro desta camada temos implementado mecanismos de comunicação *peer-to-peer* e um esquema de objetos distribuídos. O *Collaboration Controlle* permite que uma atividade seja compartilhada. Ao compartilhar uma atividade o estado de todos os serviços na atividade é sincronizado em todos os dispositivos participantes.

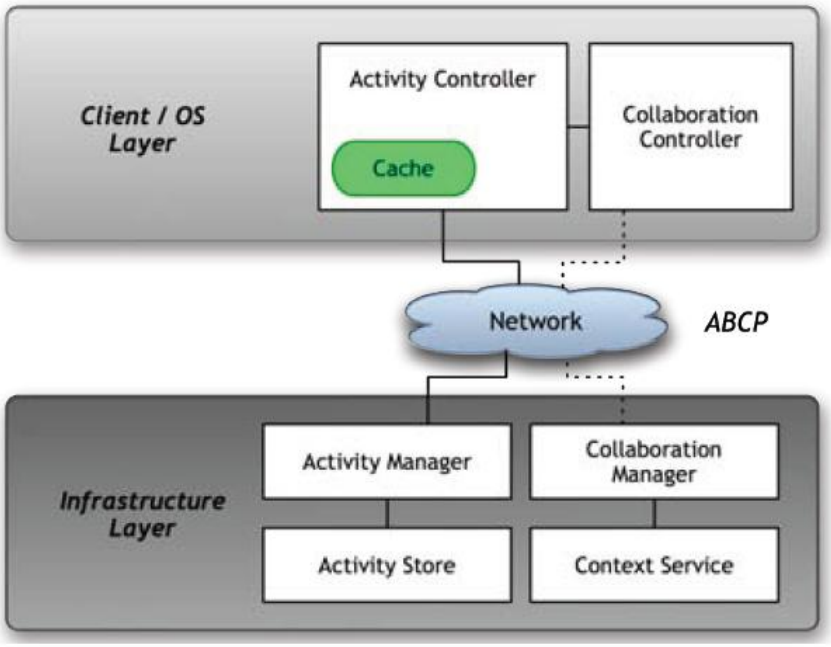

Figura 4.1: O Núcleo do Middleware ABC (PEDERSEN; MOGENSEN; BARDRAM, 2006)

Conforme Bardram (BARDRAM; CHRISTENSEN, 2007) a arquitetura do projeto ABC, exibida na figura 4.2, foi concebida com a finalidade de dar suporte à mobilidade dos usuários, cada profissional deverá carregar um dispositivo de rádio frequência, com o qual ele realiza a autenticação no sistema quando se aproxima de um dos terminais computacionais disponíveis no ambiente. Após ser detectada a presença do usuário junto ao terminal, ele confirma que deseja se logar no sistema. Dessa forma, evita-se o problema de constantes autenticações, gerado pela utilização de vários terminais no decorrer do deslocamento. Quando uma autenticação é realizada, o usuário pode executar novas tarefas ou reativar uma tarefa anteriormente suspensa. Através de *widgets* (pequenas aplicações) direcionados aos usuários, podem ser utilizados mecanismos síncronos ou assíncronos de comunicação, entre os profissionais de diferentes especialidades, permitindo o compartilhamento de informações e mecanismos colaborativos.

Usuários autenticam-se no sistema através do *Activity Bar*, a qual, quando detecta a aproximação de um usuário, a um terminal, e oferece a autenticação deste através da confirmação de que o usuário quer utilizar o sistema. O sistema procura evitar as constantes autenticações utilizando nomes de usuário e senha, presentes em sistemas mais restritivos de autenticação. É através do componente *Activity bar* que o usuário interage com o sistema de sugestão de tarefas da arquitetura (BARDRAM; CHRISTENSEN, 2007).

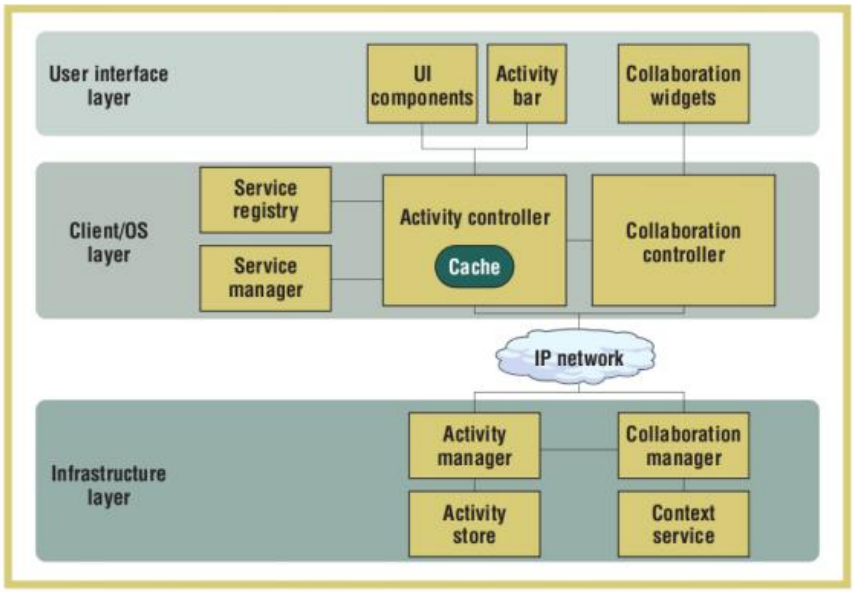

Figura 4.2: Arquitetura do Projeto ABC (BARDRAM; CHRISTENSEN, 2007)

Um característica do ABC é o *roaming* atividade. Este termo refere-se à migração das atividades de um ambiente de computação (por exemplo, um PC *Desktop*) para outro (por exemplo, a visualização na parede da sala de cirurgia, ou uma unidade móvel como o *Tablet* PC). Ao permitir a atividade tanto em *roaming*, suspensão e retomada, a computação baseada em atividades permite o médico interromper uma atividade em um dispositivo e continua-la em outro, com seu estado anterior totalmente restaurada. O sistema deve ser capaz de trazer automaticamente todos os aplicativos e os recursos associados com a atividade, aliviando assim os usuários de restauração manualmente todos os recursos e pontos de vista. Outra característica é a sensibilidade ao contexto no qual monitora continuamente e obtém informações de contexto do ambiente ubíquo assim como,

a descoberta de recursos e localização. Pode ser acessado de aplicações médicas ou pode ser configurado para notificar aplicações de acordo com o desejado, e com relação a mobilidade o projeto ABC visa atender uma área restrita como um hospital, por exemplo portanto só contempla a mobilidade local (BARDRAM, 2003) (BARDRAM, 2009) .

Em um *workshop* realizado em um hospital, a equipe de Bardram constatou que os médicos preferiram não levar um *Tablet* PC porque, era muito pesado, e grande para caber no bolso do avental branco, e difícil de lidar com ele na frente do paciente. O PDA foi considerado mais adequado, mas devido ao seu tamanho de exibição limitada, vida limitada da bateria, e falta de teclado, era percebida como útil apenas para algumas tarefas simples, como verificação do PEP ou fazer uma entrada de dados simples. Em vez disso, os clínicos preferiram usar dipositivos públicos distribuídos ao redor do hospital para trabalho clínico, e um *laptop* para um trabalho mais pessoal (BARDRAM, 2009).

A possibilidade de suspender o trabalho em um local e retomá-lo mais tarde, em outra foi muito apreciado, uma vez que permitiu um ou mais médicos colaboram para ter acesso fácil e rápido às informações.

Segundo Bardram (BARDRAM, 2009) os médicos acharam útil ser capaz de agrupar logicamente em conjunto, em uma atividade, todos os recursos e materiais relativas ao tratamento de um paciente específico, incluindo os recursos digitais, tais como registros médicos, gráficos de medicina, resultados de laboratório e imagens de Raio-X.

Os experimentos com o *framework* ABC também apontaram para uma série de desafios e áreas a serem aperfeiçoadas. Por exemplo, a falta de integração com outras aplicações clínicas hospitalares, o problema do *roaming* (migração automática) de uma aplicação de um equipamento com uma resolução, para outro com recursos diferentes de tela, a escalabilidade de muitas atividades e a falta de suporte para a colaboração baseada em funções (BARDRAM, 2009).

4.2 Projeto Awareness

O projeto Awareness (WEGDAM, 2005) propõe um *middleware* de propósitos gerais para computação ubíqua, e que utiliza a medicina como cenários de experimentos. Através desta proposta, pacientes podem ser monitorados à distância pelos profissionais de saúde.

Segundo Wegdam (WEGDAM, 2005) este projeto foi desenvolvido em colaboração entre os setores industriais e acadêmicos na Holanda, através da *University of Twente* e a *Lucent Technologies*, dentre outros institutos. O foco do projeto Awareness é a infraestrutura para a sensibilidade ao contexto que habilita a responsividade das aplicações e valida isto utilizando protótipos de aplicações móveis direcionados a área de saúde. Com o uso dessa infraestrutura de *software*, torna-se possível o monitoramento de pacientes à distância seu emprego é particularmente indicado para pacientes que estão em situação delicada de saúde.

As aplicações móveis de saúde tornam possível monitorar pacientes com doença perigosas, e até mesmo emitir alertas. Pequenos sensores médicos combinada com uma maior largura de banda e tecnologias de redes móveis confiáveis tornam possível está realidade do paciente ser controlado em qualquer lugar. Hoje em dia, em muitos pacientes, as situações são muito limitados em suas vidas e, muitas vezes tem que ficar dentro de um centro especializado de assistência médica ou em suas casas a fim de evitar manifestações imprevistas de sua doença, por exemplo ataque epiléptico ou de hipogli-

cemia em diabéticos, especialmente durante o tempo que o seu tratamento está para ser iniciado, ou ajustado. Isto lhes permite viver uma vida mais normal, e melhorar sua qualidade de vida e bem-estar. Awareness pretende investigar e demonstrar a viabilidade do tratamento de saúde móveis, ou seja, um tratamento independente de tempo e lugar utilizando um contexto de infraestrutura de serviços móveis conscientes (WEGDAM, 2005).

A arquitetura do Awareness é composta por 3 camadas: infraestrutura de rede, infraestrutura de serviços e aplicações móveis na área de saúde. Essa arquitetura pode ser visualizada na figura 4.3.

Figura 4.3: Camadas do Awareness (adaptada de (WEGDAM, 2005))

De acordo com Wegdam (WEGDAM, 2005) a primeira camada é responsável pelo acesso e uso das redes de comunicação, incluindo suporte à sensibilidade ao contexto. A camada de infraestrutura de serviços é responsável por entregar os serviços requeridos pela aplicação aos seus usuários finais. E por fim, a camada das aplicações provê sistemas voltados à área de saúde. Elas trabalham numa plataforma BAN (*Body Area Network*), que coleta dados através de sensores e os envia para os centros de tratamento e profissionais de saúde. A dinamicidade dos ambientes de computação móvel coloca novos desafios para essas aplicações em saúde, sobretudo em se tratando das condições de sinal, redução de dados (limitação de banda) e detecção automática de falhas nos sensores.

O projeto de Awareness está trabalhando em um infraestrutura que suporta sensibilidade de contexto para aplicações móveis de uma forma segura e consciente de privacidade. A infraestrutura do Awareness suporta mudanças de contexto e outras funcionalidades, como gerenciamento de identidade, gestão de utilizadores, autorização de presença e de descoberta. As funcionalidades são dissociados da lógica da aplicação e torna mais fácil desenvolver aplicativos de terceiros cientes do contexto (WEGDAM, 2005). O Awareness oferece infraestrutura de mobilidade sensível ao contexto em um ambiente de rede dinâmica, a mobilidade é suportada em duas formas: (i) a rede tem o contexto do usuário em conta, quando o controle de conectividade é fornecido ao usuário, por exemplo protocolos de roteamento dinâmico, as configurações de segurança e seleção de rede são obtidas. (ii) a rede é uma fonte de informações de contexto, para obter informações instância de presença e a largura de banda disponível (WEGDAM, 2005).

Conforme o autor Wegdam (WEGDAM, 2005) o Awareness promoverá um serviço móvel de saúde, que irá recolher sinais vitais e outras informações do paciente e usar as informações de contexto para colocar os mesmos à disposição dos profissionais de saúde que tomarão as providências necessárias.

4.3 Projeto UbiDoctor

O projeto UbiDoctor proposto por Diniz (DINIZ, 2009) da Universidade Federal de Pernambuco é uma arquitetura de serviços de *middleware* e uma aplicação protótipo que objetiva dar suporte à natureza nômade e fragmentada do trabalho médico. A aplicação protótipo é chamada UHSys (*Ubiquitous Health System*) que utiliza como cenário o ambiente UHS (*Ubiquitous Health Service*), através do qual médicos têm acesso, a qualquer hora, de qualquer lugar e usando qualquer dispositivo, a dados de pacientes contidos em PEPs. E facultado ao médico através da aplicação que ele solicite pareceres a outros colegas com relação a casos clínicos que estejam sob sua análise, bem como permite que este mesmo médico responda a solicitações de pareceres que lhes foram requisitadas.

De acordo Diniz (DINIZ, 2009) o sistema UHSys é um sistema de PEP que permite que o médico faça acesso, usando qualquer dispositivo e em qualquer hora e lugar (ou seja, de maneira ubíqua), a informações de prontuários de pacientes distribuídos entre os hospitais e unidades de saúde credenciados ao ambiente. Também é permitido que o médico faça solicitações de segunda opinião médica e analise possíveis solicitações de pareceres enviadas a ele.

A aplicação utiliza o *middleware* através das bibliotecas dos serviços de gerenciamento de contexto, de sessão e de adaptação de conteúdo. Os serviços apresentados poderão ser também usados em outros ambientes com características e requisitos similares, já que aparecem como serviços interoperáveis do *middleware*.

Segundo Diniz (DINIZ, 2009) no protótipo não foram considerados problemas relacionados à heterogeneidade da informação que está representada nas várias bases de dados nos PEPs. Embora a dificuldade de integração da informação seja conhecida, este problema ainda não foi resolvido. A aplicação possibilita ainda que o médico inicie uma sessão usando um dispositivo e no decorrer da mesma, realize a sua migração a um outro dispositivo diferente. A migração deve ser realizada com garantias de persistência nos dados e ainda, na menor fração de tempo possível, permitindo assim, um aumento de produtividade no trabalho médico. O suporte a ser dado para a realização da migração sem perdas de conteúdo ou de tempo é oferecido pelo serviço de gerenciamento de sessão e de contexto. É necessário ainda adaptar o conteúdo a ser mostrado ao médico, dependendo do dispositivo que ele utilize, e para isto, o UHSys faz uso do serviço de gerenciamento de contexto e do serviço de adaptação de conteúdo do UbiDoctor.

A figura 4.4 apresenta a arquitetura do UHSys. Na base da arquitetura estão os serviços do *middleware* UbiDoctor. O serviço de mais baixo nível é o de gerenciamento de contexto, que presta suporte aos serviços de gerenciamento de sessão e adaptação de conteúdo. Acima do *middleware*, existem os servidores de aplicação dos PEPs. Estes servidores estão distribuídos no ambiente e podem ser acessados através de um servidor Web (*World Wide Web*), que no caso específico da prototipação deste trabalho, foi utilizado o servidor de *Web Apache TomCat* (TOMCAT APACHE, 2009). Este conjunto de componentes constituem o módulo *back-end* do cenário, ou seja, a parte referente aos serviços (de *middleware* e de aplicação) oferecidos aos usuários.

Conforme Diniz (DINIZ, 2009) no *front-end* do sistema, existem usuários que acessam o ambiente através de *Smartphones*, fazendo uso de uma interface desenvolvida em JME (*Java Micro Edition*), ou usuários que utilizam a interface *web-browser* de seus computadores pessoais e PDAs. Através do *front-end*, um médico poderá consultar um PEP e acrescentar informações demográficas, clínicas, exames, prescrições, dentre outros

dados. Os dados do paciente podem estar distribuídos na rede UHS e o médico poderá ter acesso a eles de seu consultório, de sua casa, em locais de lazer ou em trânsito, usando para isto, um dispositivos móvel ou não e com várias possibilidades de redes de acesso.

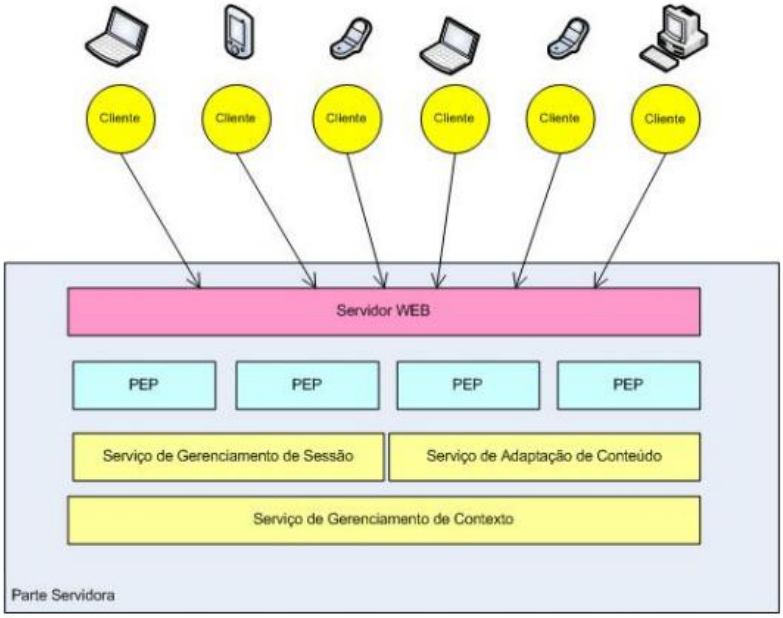

Figura 4.4: Arquitetura do Protótipo UHSys (DINIZ, 2009)

O UbiDoctor propõe-se utilizar uma solução baseada em *middleware*, na qual os serviços propostos localizam-se na subcamada de serviços comuns conforme definido por Schmidt (SCHMIDT, 2000), podendo ser re-usados em outros ambientes com características e requisitos similares. Este conjunto de serviços que dão suporte ao ambiente UHS (*Ubiquitous Health Service*) provê a realização das seguintes tarefas (DINIZ, 2009):

- suporte à adaptação de conteúdo em ambientes sensíveis a contexto, considerando as configurações heterogêneas dos dispositivos e a abrangência da rede;
- suporte ao gerenciamento de sessão, com objetivo de proporcionar a migração de aplicações, manutenção e persistência de sessões;
- suporte à execução de serviços de PEP, distribuídos nos hospitais da rede ou postos de saúde, bem como, a integração entre todos os seus usuários.

No ambiente UHS, o tratamento a ser dado a essas situações pode ser obtido através do apoio dos serviços de gerenciamento de contexto, de gerenciamento de sessão e adaptação de conteúdo providos pelo *middleware* UbiDoctor. A arquitetura de serviços do UbiDoctor é apresentada na figura 4.5.

Os serviços de adaptação de conteúdo e gerenciamento de contexto tentam minimizar o problema dos diferentes tamanhos de telas e configurações dos dispositivos envolvidos no cenário. Os serviços de gerenciamento de contexto e de gerenciamento de sessão também tratam as possíveis interrupções durante a realização da migração de aplicações. Por fim, o serviço de gerenciamento de sessão preocupa-se ainda com os possíveis atrasos envolvidos no processo de migração da mesma.

A sensibilidade ao contexto é provida pelo serviço de gerenciamento de contexto que trabalha seguindo a abordagem *publish/subscribe* relacionando os monitores e o gerenciador de contexto. No caso da localização, percebe-se que a mudança da variável de contexto deve ser percebida de forma espontânea (*ad hoc*), sem necessariamente ter uma iniciativa do médico em informar que está entrando ou saindo de um determinado hospital ou postos de saúde. O próprio monitor deverá perceber essa mudança e informar ao gerenciador. Como o monitor é um publicador de eventos (*publisher*) associados à mudança da variável de localização e o gerenciador de contexto é um assinante (*subscriber*), este último será avisado desta mudança.

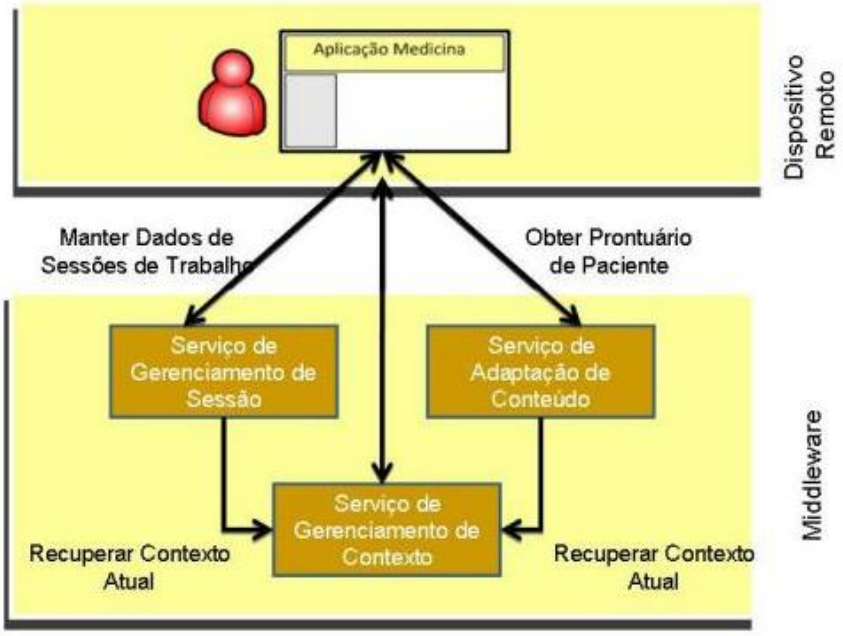

Figura 4.5: Arquitetura de Serviços UbiDoctor (DINIZ, 2009)

Como os serviços propostos pelo UbiDoctor operam na camada de serviços comuns, é necessário que as subcamadas inferiores do *middleware*, segundo o modelo de Schmidt (SCHMIDT, 2000), dêem suporte à inferência de localização e informações de contexto.

Segundo Diniz (DINIZ, 2009) para avaliação do conjunto de serviços propostos pelo UbiDoctor foram elaboradas duas análises: a primeira delas foi uma análise quantitativa, e a segunda, uma análise qualitativa referente ao uso dos serviços a partir do aplicativo UHSys. Através da pesquisa quantitativa foi possível observar qual era o impacto do tempo de migração de sessões considerando-se três experimentos. Para o primeiro experimento, foi considerado o uso de um *Desktop* para suspender e outro para resgatar a sessão. No segundo experimento, considerou-se um *Smartphone* como dispositivo de origem, usado portanto para suspender a sessão e um *Desktop* para resgatá-la. Por fim, no terceiro experimento, foi utilizado um *Desktop* para suspender a sessão e um *Smartphone* para resgatá-la. Para cada um destes experimentos, o tempo de migração foi observado para duas atividade avaliadas. O objetivo foi verificar se o impacto do uso dos serviços seria baixo em relação aos 51 segundos obtidos pelo estudo de observação de Tentori e Favela (TENTORI; FAVELA, 2008), no qual diz que um profissional de saúde leva em média 51 segundos na mudança de uma atividade para outra.

Conforme os experimentos realizados, o tempo de migração total observado é mínimo, de modo a representar um baixo impacto no tempo observado entre duas atividades executadas sucessivamente. Isto significa que, por exemplo, um médico quando está acessando dados de um PEP em um *Notebook* e é interrompido por um chamado de urgência no hospital, poderá migrar sua sessão e ter a possibilidade de concluí-la até chegar ao local de sua próxima atividade. Isto pode permitir um aumento na produtividade durante a realização de suas atividades de rotina.

De acordo com Diniz (DINIZ, 2009) foi possível verificar situações que os médico solicitariam a migração de aplicações e se eles julgarem que o processo aconteceria em um tempo razoável.

Portanto essa conclusão reforça a hipótese de que se o tempo de migração de sessão foi pequeno, ainda que seja necessário adaptar conteúdo, os médicos acreditariam na possibilidade de ganho de produtividade.

4.4 Projeto ClinicSpace

O projeto ClinicSpace, desenvolvido pelo grupo de pesquisa GMob da Universidade Federal de Santa Maria, visa a utilização de tecnologias providas pela computação ubíqua para o auxílio aos profissionais clínicos na execução de suas tarefas nos ambientes hospitalares. Ressalta-se que, mesmo o sistema ubíquo devendo ser pró-ativo (agir em nome do usuário), esse deve levar em conta que o usuário deseja manter a sua forma de executar o seu trabalho. Dessa forma, objetiva-se fornecer mecanismos que permitam aos usuários interagir e personalizar o sistema, adequando-o melhor a sua forma de executar as tarefas. Argumenta-se que a personalização do modo como cada um executa suas atividades, aliado a soluções da ubiquidade, pode levar a diminuir a rejeição dos sistemas computacionais dentro do ambiente hospitalar (SILVA, 2010).

O projeto propõe a definição de uma ferramenta piloto, conforme pode ser visualizado na figura 4.6, na forma de uma interface, para a personalização de tarefas, realizada pelo próprio usuário (médicos) com vistas ao gerenciamento de suas atividades clínicas, relativas a diagnóstico, tratamento, laboratorial e administrativas. O sistema proposto é visto como um assistente, auxiliando o médico a executar as tarefas que compõem suas atividades profissionais diárias.

Segundo Silva (SILVA, 2010) a arquitetura para a programação e gerenciamento personalizado das tarefas, mostrada na figura 4.7, foi organizada em níveis que refletem as visões do sistema: (i) nível superior, é composto pelo usuário final (médico) que interage com a ferramenta para (re)programar suas tarefas que executarão num ambiente pervasivo; (ii) nível intermediário, é composto pelo mapeamento entre tarefas (definidas pelo usuário) e subtarefas (aplicações pervasivas) e pelo gerenciamento de ambas; (iii) nível inferior, é composto pelo conjunto de serviços do *middleware* de gerenciamento do ambiente pervasivo e de suporte à execução das aplicações pervasivas. *Middleware* empregado no projeto ClinicSpace é o EXEHDA (YAMIN, 2005).

A execução das tarefas é gerenciada pelo *middleware* EXEHDA (YAMIN, 2005). Porém, como o EXEHDA não foi desenvolvido para ser orientado a tarefas, a introdução do conceito de orientação a tarefas (*activity-driven* ou *task-oriented computing*) exigiu a inserção de um novo subsistema no *middleware*, estendendo suas funcionalidades. O novo SGDT (Subsistema de Gerenciamento Distribuído de Tarefas) tem a função de fazer a ponte entre tarefas e aplicações pervasivas, conforme definidas na arquitetura do *mid-*

Figura 4.6: Protótipo do ClinicSpace (SILVA, 2010)

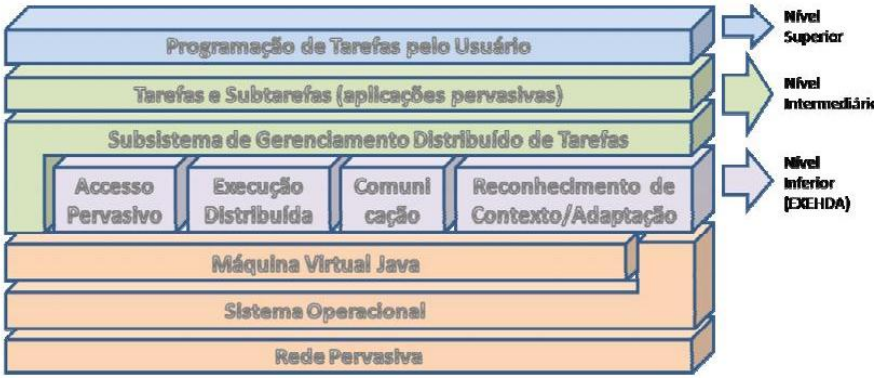

Figura 4.7: Arquitetura para Programação e Gerenciamento de Tarefas (SILVA, 2010)

dleware, auxiliando no processo de conversão de tarefas-aplicações. Portanto, o SGDT é responsável por gerenciar, em alto nível, as tarefas, delegando o gerenciamento das subtarefas (aplicações pervasivas) para os subsistemas atuais do EXEHDA (SILVA, 2010).

No entanto a arquitetura do EXEHDA (YAMIN, 2005) provém de forma nativa vários serviços dentre eles: sensibilidade de contexto (fornecido na forma de *framework*), monitoramento, distribuição, adaptação, entre outros.

Os eventos programáveis pelo usuário (intervenção), e uma funcionalidade do EXEHDA (YAMIN, 2005), responsável por receber os parâmetros informados pelo usuário e realizar as operações necessárias para sua efetivação na arquitetura.

A interveção no ClinicSpace ocorre através dos atuadores que são controlados pelo SGCT, utilizando os serviços base de monitoramento de sensores e, contexto do EXEHDA (YAMIN, 2005). No momento de sua construção, é especificado um ou mais gatilhos monitorados pelo *middleware* e que, quando atingidos, são responsáveis por executar a ação prevista pelo atuador. Salienta-se que os atuadores são entidades construídas com um propósito específico, onde é permitido ao usuário ligá-los às tarefas que desejar, personalizando o valor dos elementos de disparo. O agendamento é um exemplo de atuador, onde seu gatilho principal é constituído pela data e hora.

Para o desenvolvimento do projeto ClinicSpace, buscou-se o suporte fornecido por guias clínicos e estudos realizados junto a profissionais da saúde para nortear o desenvolvimento do sistema. As tarefas do usuário no caso profissionais de saúde, são executadas em um ambiente dinâmico, o que implica na necessidade da arquitetura oferecer suporte a um conjunto de características, necessárias as tarefas, sendo (RIZZETTI, 2010):

- decomposição, recombinação e reuso - tarefas são formadas por subtarefas reusáveis. Essas subtarefas podem ser recombinadas de diversos modos para formar diferentes tarefas; como diferentes tarefas possuem subtarefas em comum, elas podem ser reutilizadas para desenvolver novas tarefas e assim reaproveitar a programação das subtarefas já implementadas;

- interrupção (preemptáveis) - as tarefas podem ser suspensas e retomadas mais tarde (semelhante ao modelo de co-rotina);

- mobilidade e adaptabilidade - podem migrar e se adaptar para acompanhar o usuário em movimento (semântica siga-me);

- contextualização - as tarefas são associadas a um contexto, o que pode ocorrer dinamicamente. O contexto, assim como as próprias tarefas, são elementos modularizados, programáveis pelo usuário, capazes de proporcionar suporte a recombinação e reuso entre diferentes tarefas;

- semântico - visando prover essa automatização, torna-se necessário que o sistema de contexto conheça quais são os dados de que a aplicação necessita, para que possa monitorá-los. Na maioria das vezes, não basta efetuar o monitoramento dos dados brutos de contexto, é necessário agregar uma significação semântica a eles, baseado em faixas de valores válidos, nomenclaturas clínicas ou outras informações que agregam valor semântico à informação;

- personalização - tem por finalidade permitir a personalização do contexto monitorado, na forma de mecanismos para entrada implícita de dados e mecanismos

pró-ativos para execução de ações baseado na programação efetuada pelo usuário-final.

Na interação do usuário com o ambiente pervasivo, como por exemplo, em um hospital, mais especificamente na movimentação pelo ambiente, existem três casos que são considerados (RIZZETTI, 2010):

i. O usuário pode se locomover dentro do ambiente hospitalar ou fora dele, portando um dispositivo móvel. Nesse caso, não há desconexão, e a sessão do usuário permanece ativa durante o deslocamento.

ii. O usuário ao sair do dispositivo onde estava trabalhando, o sistema encerra a sessão aberta, interrompendo e armazenando as tarefas que estavam em execução. Ao aproximar-se de outro dispositivo, seu login é habilitado nesse dispositivo, bastando o usuário logar-se para que sua sessão seja restaurada, e suas tarefas sejam mostradas na tela do dispositivo.

iii. O usuário pode migrar de dispositivo. Nesse caso, ele pode deslogar-se do dispositivo que está usando e logar-se no que irá utilizar. Ou, simplesmente, logar-se no dispositivo que irá utilizar, fazendo com que o sistema encerre sua sessão no dispositivo que estava sendo utilizado. Em ambos os casos, as tarefas em execução são interrompidas, armazenadas e disponibilizadas para continuação na tela do dispositivo no qual o usuário se logou.

Conforme Rizzetti (RIZZETTI, 2010) o projeto ClinicSpace, conforme discutido anteriormente, visa construir uma arquitetura para facilitar a execução de tarefas computacionais no meio clínico. Para isso, utiliza-se o paradigma de computação orientada à tarefa (*Task Computing*), com forte foco na questão de personalização de tarefas pelo usuário médico. Um dos aspectos fundamentais do ambiente pervasivo é fornecer a capacidade do ambiente atuar como fornecedor de informações para a arquitetura, liberando o usuário da entrada explícita de dados tanto quanto possível, através da obtenção destes por um sistema de monitoramento, o qual pode ser implementado através de sensores que permitem ao sistema obter os dados de que necessita e gerar possíveis notificações aos usuários.

De acordo com Silva (SILVA, 2010) testes de campo não poderão ser realizados devido a disponibilização do protótipo da arquitetura de gerenciamento e execução das tarefas, o projeto ClinicSpace, até o momento não está totalmente concluído.

4.5 Síntese dos Trabalhos Relacionados

Considerando aspectos de afinidade foram selecionados quatro trabalhos relacionados, (ABC, Awareness, UbiDoctor e ClinicSpace) respectivamente. Observa-se que os quatro contemplam soluções de *middleware* e serviços para ambientes de medicina ubíqua. Alguns, utilizam como cenário a rotina diária de um hospital, inclusive utilizando sensores, em alguns casos, equipamentos ubíquos espalhados em alguns setores hospitalares, outros utilizam como cenário o monitoramento de pacientes à distância. Também estão associados a interações entre médicos e pacientes ou entre grupos de médicos, de modo a prover, em alguns casos, a participação do paciente no tratamento.

Na tabela 4.1 e verificado se determinadas características estão ou não presentes nos trabalhos relacionados os que tiverem assinalados com “X” indica que a característica está presente:

- mobilidade remota: os trabalhos relacionados contemplam a mobilidade remota, ou seja, atende uma área extensa, como por exemplo, uma região metropolitana, ou só atendem a mobilidade local que consiste em uma área restrita, como por exemplo, um hospital;

- adaptação de conteúdo: preocupa-se com o desconforto no uso de alguns dispositivos portáteis permitido a migração das sessões para dispositivos mais adequados. Para oferecer suporte a essa multiplataforma, existe o serviço de adaptação de conteúdo;

- sensibilidade ao contexto: capacidade de um sistema computacional perceber características contextuais de seu interesse, é um requisito chave para permitir a adaptação em resposta às mudanças ambientais. Afim de coletar informações sobre o ambiente onde está sendo utilizado, existe o serviço de sensibilidade ao contexto;

- processamento semântico: utiliza processamento semântico em mecanismo de sensibilidade ao contexto, desde a sua aquisição, processamento e distribuição das informações contextuais, direcionado a computação ubíqua;

- intervenção no ambiente ubíquo: consiste em uma funcionalidade, que permite realizar ações sobre o meio físico. No entanto, estas ações podem ser realizadas de acordo com políticas e parâmetros como: tipo de ação (ex. ligar, desligar dispositivos), tempo de duração (ex. ligar por 30 segundos), potência (ex. ligar na velocidade máxima), entre outras variações específicas de cada atuador. Nesta perspectiva, o atuador participa das rotinas da arquitetura;

- notificação automática: consiste na coleta automatizada de dados coletados e seu respectivo processamento. Em função do processamento dos dados no mecanismo de sensibilidade ao contexto podem ser emitidas notificações aos profissionais envolvidos;

- descoberta de recursos: é possível haver descoberta de recursos computacionais no ambiente ubíquo e quais atendem as necessidades da aplicação para que possam ser alocados e desalocados com menor interferência possível do usuário;

- personalização de regras: possibilidade de alteração de regras de monitoramento em tempo de execução. Entende-se por tempo de execução a flexibilidade da arquitetura de software em reconfigurar dinamicamente as regras para manipulação e processamento das informações contextuais.

Conforme pode ser observado na tabela 4.1 alguns trabalhos relacionados analisados possuem características em comum, contudo com a pesquisa destes foi possível conhecer um pouco sobre a área de medicina ubíqua seus objetivos e desafios a serem vencidos. O estudo dos trabalhos relacionados foi fundamental para elaboração desta dissertação de mestrado.

Tabela 4.1: Síntese dos Trabalhos Relacionados

Características	**ABC**	**Awarenes**	**UbiDoctor**	**ClinicSpace**
Mobilidade Remota		X	X	X
Adaptação de Conteúdo	X		X	X
Sensibilidade ao Contexto	X	X	X	X
Processamento Semântico				X
Intervenção no Ambiente Ubíquo				X
Notificação Automática	X	X		X
Descoberta de Recursos	X	X		
Personalização de Regras				X

4.6 Considerações sobre o Capítulo

Este capítulo apresentou alguns projetos relevantes na área de medicina ubíqua apontando suas funcionalidades e benefícios, assim como restrições a serem vencidas para uma melhor eficácia de seus serviços. Por fim foi apresentado uma tabela comparativa abrangendo os projetos abordados neste capítulo. No próximo capítulo é apresentado os fundamentos utilizados para a arquitetura do uMED.

5 UMED: FUNDAMENTOS

Neste capítulo, são discutidos os principais fundamentos relacionados ao uMED. Dentre os quais destaca-se o projeto PERTMED o qual este trabalho esta inserido, o *middleware* EXEHDA, sua organização e subsistemas. Assim como o mecanismo de sensibilidade ao contexto denominado de EXEHDA-SS, o qual é parte fundamental da arquitetura proposta para o uMED. Os subcapítulos descritos a seguir, constituem os principais fundamentos utilizados na proposição do uMED.

5.1 Projeto PERTMED

O projeto PERTMED (PERTMED, 2007) está sendo desenvolvido por universidades do sul do Brasil e conta com a colaboração de equipes médicas ligadas a essas universidades.

O sistema de saúde do futuro prevê o uso de tecnologias da computação ubíqua formando um espaço inteligente (reativo e pró-ativo), onde dispositivos móveis e fixos estão integrados ao ambiente físico (objetos) visando captar informações do meio e transmitir as alterações detectadas para sistemas de gerenciamento de informações, os quais tomarão decisões e adaptar-se-ão as situações detectadas (computação consciente de contexto) (PERTMED, 2007).

Neste momento, em termos de pesquisa e inovações, a computação ubíqua na saúde está sendo conduzida sob duas perspectivas (PERTMED, 2007): (i) uso de tecnologias da computação ubíqua para criar um hospital virtual; (ii) disponibilização das informações relativas a saúde em todo lugar, a qualquer tempo, usando diferentes dispositivos de acesso pertencentes ao próprio paciente/médico ou dispositivo no ambiente.

Conforme o projeto PERTMED (PERTMED, 2007) no Brasil, os trabalhos tendo a saúde e computação ubíqua como tema estão em fase inicial.

Usando a experiência do GMob (Grupo de Pesquisa em Sistemas Móveis e Pervasivos), da UFSM (Universidade Federal de Santa Maria), do G3PD da UFPeL (Universidade Federal de Pelotas) e da UCPeL (Universidade Católica de Pelotas) e a qualificação da equipe médica dos hospitais universitários dessas universidades, propõem-se uma inovação nos sistemas de comunicação e informação através do uso de estratégias usadas na computação ubíqua para inserir aspectos importantes ainda não presentes nos sistemas de informação de saúde do país. A premissa é tornar a informação disponível aonde (lugar) ela é necessária de forma contextualizada. A disponibilidade da informação é fundamental para qualidade da tomada de decisão.

O projeto PERTMED (PERTMED, 2007) propõe fazer a ligação entre os sistemas

automatizados existentes (registro de pacientes, exames laboratoriais, entre outros) e o médico no local onde ele se encontra (regiões remotas ou em trânsito, por exemplo). Sendo assim, elimina-se a exigência da conexão a uma rede fixa e com um computador pessoal na área do hospital para ter acesso às informações do paciente.

O projeto PERTMED (PERTMED, 2007) usa as funcionalidades e capacidades fornecidas pelos telefones celulares e *Smartphones* que tornam informações do paciente disponível para o médico e demais profissionais de saúde com direito de acesso, em qualquer lugar que esteja necessitando dessas informações para tomada de decisão. No momento que este solicita a informação, ela é acessada e enviada para o dispositivo móvel em uso pelo médico ou por outro profissional de saúde no momento (por exemplo, *Smartphone*) e adaptada ao dispositivo que o médico utiliza.

Logo, a distância do médico e demais profissionais de saúde em relação ao sistema de equipamentos que armazenam as informações sobre o paciente é irrelevante, tornando o sistema de informação mais flexível e de acordo com a liberdade de movimentação requerida pelos profissionais da saúde (PERTMED, 2007).

A *pervasive healthcare* questão central do PERTMED está sendo considerada a próxima etapa da *Web-based Healthcare Computing* que oferece vantagens competitivas aos provedores de serviço de saúde; em particular, aumenta a eficiência do serviço, a qualidade e melhora o gerenciamento da relação com o paciente (VARSHNEY, 2003). Este novo sistema de saúde também prevê uma visão de hospital virtual, o qual estende-se para a casa dos pacientes ou lugares onde eles se encontram, onde sensores/dispositivos monitoram as condições ambientais e do paciente e comunicam-se, via rede sem fio, com as centrais médicas para tomada de decisões e ações pertinentes. Experiências nesse sentido estão sendo conduzidas por alguns projetos de pesquisa europeu, como o do *Centre of Pervasive Healthcare* na Dinamarca que desenvolve o projeto *Hospital of the Future* (BARDRAM; BOSSEN, 2005).

Por conseguinte, a computação ubíqua atingirá uma volumoso potencial de aplicabilidade na área da saúde. O projeto PERTMED (PERTMED, 2007) tem como motivações contribuir para que sejam superados alguns desafios de área de saúde dentre eles destaca-se:

- dificuldade de acesso a serviços especializados em regiões remotas ou carentes;
- custo elevado para o transporte de pacientes, especialmente de áreas pobres e rurais;
- interrupções e atrasos na sequência do tratamento.

Segundo o projeto PERTMED (PERTMED, 2007) a questão que permeia esses três problemas é o acesso a informação de onde ela é gerada para onde ela é necessária, em tempo razoável e compatível com a gravidade da situação sendo tratada. A rapidez da decisão médica depende da pronta disponibilidade destas informações, sendo este o fator decisivo para a qualidade dos serviços prestados. Acesso à informação pode ser usado para substituir o transporte, por exemplo, um "paciente virtual" (formado por um conjunto de informações sobre seu estado de saúde, obtidos através de sensores) pode ser acessado a longas distâncias por um especialista remoto.

Com a introdução da computação ubíqua na área da saúde, objetiva-se dentre outros aspectos, prover uma contribuição no sentido de superação de desigualdades regionais e sócioeconômicas, relativas ao acesso às informações dos sistemas de saúde (PERTMED, 2007).

Em termos científicos, o foco do projeto é avaliar o potencial de aplicabilidade de algumas estratégias e tecnologias usadas na computação ubíqua para os sistemas de saúde: (i) credenciais associadas aos modernos códigos de barras multimensionais (*tags*) e (ii) mecanismos de disseminação pervasiva de dados no ambiente *Internet* móvel (especialmente, *Smartphones*), às informações sobre o PEP, respeitadas as restrições de segurança e privacidade.

O PERTMED (PERTMED, 2007) irá fazer a ligação entre os sistemas automatizados existentes e o médico e demais profissionais de saúde, seja lá qual for o local onde se encontram. Com isso, atende-se a necessidade de flexibilidade e liberdade de movimentação do médico e demais profissionais no atendimento aos pacientes. Também, haverá uma melhoria na agilidade dos serviços uma vez que a informação poderá ser obtida assim que for gerada (resultado de um exame laboratorial, por exemplo). Agilidade promoverá entre outras a redução de filas de atendimento, e do tempo de retorno dos resultados.

Segundo o projeto PERTMED (PERTMED, 2007) outra possibilidades que a tecnologia permite e que pode ser avaliada é a de orientação simples (mensagens) serem enviadas aos pacientes, por exemplo via celular ou *Smartphone*, a partir da decisão do médico. O uso da tecnologia para comunicação médico-paciente irá beneficiar o sistema de saúde, em termos de agilidade no atendimento, no qual contribui para uma melhora na qualidade do serviço prestado.

O acesso a redes de transmissão (banda larga, GSM entre outras) e a computadores portáteis não são a realidade em muitas regiões e hospitais. Porém, um pequeno computador - o telefone celular - está amplamente difundido e está mais presente nas casas do que computadores pessoais com acesso à *Internet*, segundo dados oficiais (INFOEXAME, 2007). Logo, pode-se tirar vantagens da disponibilidade de comunicação via telefonia móvel, a qual permite independência de lugar e tempo, e começar a explorar seu uso na área de saúde, com o desenvolvimento de aplicações viáveis, simples e eficazes.

O custo de transmissões de dados digitais está ficando mais acessível conforme a revista Infoexame (INFOEXAME, 2007). A tendência é de declínio dos preços a medida que aumenta o uso de aplicações além das conversas telefônicas. As operadoras de telefonia móvel podem oferecer planos especiais a um custo razoável para o uso de aplicações voltadas a área de saúde.

Espera-se que, com o início de cooperação entre as equipes de computação e as equipes médicas das instituições, crie-se uma prática de transferência de tecnologia da pesquisa em computação ubíqua para a saúde, fazendo aplicações reais que usam os conhecimentos gerados como resultado da pesquisa como ocorre em países desenvolvidos (PERTMED, 2007).

Para o desenvolvimento do sistema está sendo utilizado métodos, técnicas e ferramentas de análise e projetos orientado a objetos. Particularmente, usam-se padrões de projetos e diagramas UML (*Unified Modeling Language*) que auxiliam na modelagem do sistema, os quais facilitam futuras alterações/evoluções.

A linguagem utilizada para o desenvolvimento é a plataforma Java. Está foi escolhida pela ampla aceitação, facilidades fornecidas para projetos na área de mobilidade e Web, e pela portabilidade o que facilita a programação de PDAs, telefones celulares, *Smartphones* entre outros.

O projeto PERTMED (PERTMED, 2007) prevê o uso do EXEHDA como *middleware* direcionado a computação ubíqua.

5.2 Middleware EXEHDA

Este seção registra de modo resumido a revisão feita sobre o *middleware* EXEHDA (YAMIN, 2004) e seus subsistemas.

O EXEHDA (YAMIN, 2004) é um *middleware* adaptativo ao contexto e baseado em serviços que visa criar e gerenciar um ambiente ubíquo, bem como promover a execução, sob este ambiente, das aplicações que expressam a semântica siga-me. Estas aplicações são distribuídas, móveis e adaptativas ao contexto em que seu processamento ocorre, estando disponíveis a partir de qualquer lugar, todo o tempo (LOPES; PILLA; YAMIN, 2007).

5.2.1 Aspectos Funcionais e Arquiteturais

De acordo com Yamin (YAMIN, 2004) o EXEHDA é um *middleware* que compõem o projeto ISAM (ISAM, 2009), dirigido para às aplicações distribuídas, móveis e sensíveis ao contexto da computação ubíqua. A seguir são apresentadas premissas a serem alcançadas na concepção do EXEHDA.

5.2.1.1 Premissas de Pesquisa do EXEHDA

As principais premissas de pesquisa do *middleware* EXEHDA, são apresentadas considerando, a dinamicidade e heterogeneidade do ambiente de processamento, o controle da adaptação, o suporte às mobilidades lógica e física e o suporte da semântica siga-me.

Dinamicidade e Heterogeneidade do Ambiente de Processamento

A movimentação do usuário (mobilidade física), aspecto típico da computação ubíqua, determina a execução de aplicações a partir de diferentes dispositivos e/ou pontos da rede global, nos quais a oferta e/ou disponibilidade dos recursos computacionais é variável. Disto decorre:

- alta flutuação na banda passante disponível para as comunicações;
- equipamentos de usuários com grandes diferenças nos atributos de *hardware* e sistema operacional;
- diferentes infraestruturas para conexão à rede global.

Suporte a Sensibilidade ao Contexto

A gerência da adaptação pode surgir no limite de dois extremos: (i) no primeiro, denominado *laissez-faire*, a aplicação é responsável por toda a adaptação que será realizada; por sua vez, no segundo extremo, (ii) denominado *application-transparent*, o sistema é encarregado de gerenciar toda a adaptação que vier a ocorrer. No entanto, nenhuma dessas estratégias pode ser considerada a melhor. Uma estratégia diferente pode ser requerida para cada circunstância e/ou aplicação. Há situações, por exemplo, onde o código fonte da aplicação não está disponível e a estratégia a ser utilizada deve ser a *application-transparent*. Em outros casos, pode ser mais conveniente incluir apenas na aplicação os mecanismos adaptativos, sem envolver o ambiente de execução. Logo, a proposta para o EXEHDA é modelar um *middleware* que possibilite uma estratégia colaborativa com a aplicação nos procedimentos de adaptação. Deste modo, considerando a

natureza da aplicação, o programador poderá definir a distribuição de responsabilidades entre a aplicação e o *middleware* no processo de adaptação.

Suporte às Mobilidades Lógica e Física

Genericamente computação móvel se refere a um cenário onde todos, ou alguns nodos que tomam parte no processamento, são móveis. Desta definição podem derivar diferentes interpretações. Em um extremo, a mobilidade leva em conta as necessidades dos usuários nômades, isto é, usuários cuja conexão na rede ocorre de posições arbitrárias e que não ficam permanentemente conectados. Em outro extremo, estão os usuários móveis, os quais retêm a conectividade durante o deslocamento, tipicamente explorando conexões sem fio. Sendo assim, a computação móvel é caracterizada por três propriedades: mobilidade, portabilidade e conectividade.

Na proposta do EXEHDA estas propriedades oferecem dois desafios de pesquisa: (i) os segmentos sem fio da rede global levantam novas condições operacionais, entre as quais se destaca a comunicação intermitente. A ocorrência de desconexões de nodos no ambiente móvel, sejam estas voluntárias ou não, é mais uma regra do que uma exceção; (ii) a natureza dinâmica do deslocamento do *hardware* e do *software* na rede global introduz questões relativas tanto à identificação física dos nodos quanto à localização dos componentes de *software* que migram.

Estes desafios apontam para a necessidade de mecanismos dinâmicos que realizem o mapeamento dos componentes móveis, de modo a viabilizar sua localização e permitir a interação com os mesmos. Logo, o *middleware* para suporte à computação ubíqua deve levar em conta essas limitações, de modo que as aplicações não percam sua consistência quando um componente de *software* migrar entre nodos da rede global, ou quando um nodo temporariamente não estiver disponível por estar desligado ou sem conexão, ou ainda trocar sua célula de execução em função do deslocamento.

O *middleware* deverá viabilizar a semântica siga-me das aplicações ubíquas. A localização é um fator determinante para os sistemas com mobilidade, pois a localidade influi significativamente no contexto disponibilizado para os mecanismos de adaptação. O contexto representa uma abstração peculiar da computação ubíqua, e inclui informações sobre recursos, serviços e outros componentes do meio físico de execução. Nesta perspectiva, o ambiente de execução deve fornecer informações de contexto que extrapolam a localização onde o componente móvel da aplicação se encontra.

Suporte a Semântica Siga-me

Segundo Yamin (YAMIN, 2004) proporcionar suporte à semântica siga-me é uma das contribuições centrais do EXEHDA ao Projeto ISAM. No EXEHDA, este suporte é construído pela agregação de funcionalidades relativas ao reconhecimento de contexto, ao acesso pervasivo e à comunicação. Como estratégia para tratamento da complexidade associada ao suporte da semântica siga-me, no EXEHDA é adotada a decomposição das funcionalidades de mais alto nível, recursivamente, em funcionalidades mais básicas. Nesta ótica , no EXEHDA o reconhecimento de contexto está relacionado com dois mecanismos: (i) um de monitoração que permite inferir sobre o estado atual dos recursos e das aplicações, e (ii) outro que pode promover adaptações funcionais e não funcionais, tendo em vista o contexto monitorado.

Entende-se por adaptação não-funcional a capacidade do sistema atuar sobre a localização física dos componentes das aplicações, seja no momento de uma instanciação

do componente, seja, posteriormente, via migração do mesmo. Ambas operações demandam a existência de mecanismo para instalação sob demanda do código, assim como mecanismos para descoberta e alocação dinâmicas de recursos e acompanhamento de seu estado. Já a adaptação funcional consiste na capacidade do sistema atuar sobre a seleção da implementação do componente a ser utilizado em um determinado contexto de execução. Novamente surge a necessidade do suporte à instalação de código sob demanda. A funcionalidade da instalação sob demanda implica que o código a ser instalado esteja disponível em todos os dispositivos nos quais este venha a ser necessário. Considerando as dimensões do ambiente ubíquo, é impraticável manter a cópia de todos os possíveis códigos em todos os eventuais dispositivos. Procede daí a necessidade de um mecanismo que disponibilize acesso ubíquo ao repositório de código, mecanismo este, que deve considerar fortemente o aspecto escalabilidade. O aspecto de mobilidade, tanto dos componentes das aplicações quanto do usuário, inerente à semântica siga-me, faz-se necessária uma estratégia de comunicação caracterizada pelo desacoplamento espacial e temporal.

5.2.1.2 Organização do EXEHDA

O *middleware* EXEHDA tem como objetivo definir a arquitetura para um ambiente de execução destinado às aplicações da computação ubíqua, no qual as condições de contexto são pró-ativamente monitoradas, e o suporte à execução deve permitir que tanto a aplicação como ele próprio utilizem estas informações na gerência da adaptação de seus aspectos funcionais e não-funcionais. Entende-se por adaptação funcional aquela que implica a modificação do código sendo executado. Por sua vez, adaptação não funcional, é aquela que atua sobre a gerência da execução distribuída. Também a premissa siga-me das aplicações ubíquas deverá ser suportada, garantindo a execução da aplicação do usuário em qualquer tempo, lugar e dispositivo (YAMIN, 2004).

As aplicações alvo são distribuídas, adaptativas ao contexto em que executam e compreendem a mobilidade lógica e a física. Na perspectiva do EXEHDA, entende-se por mobilidade lógica a movimentação entre dispositivos de artefatos de *software* e seu contexto, e por mobilidade física o deslocamento do usuário, portando ou não seu dispositivo (ISAM, 2009).

Arquitetura de Software

A arquitetura de *software* do *middleware* EXEHDA apresentada na figura 5.1. Os principais requisitos que o EXEHDA deve atender são: (i) gerenciar tanto aspectos não funcionais como funcionais da aplicação, e de modo independente, (ii) dar suporte à adaptação dinâmica de aplicações; (iii) disponibilizar mecanismos para obter e tratar informações de contexto; (iv) empregar informações de contexto na tomada de decisões, (iv) decidir as ações adaptativas de forma colaborativa com a aplicação e (v)disponibilizar a semântica siga-me, permitindo ao usuário o iniciar as aplicações e o acesso a dados a partir de qualquer lugar, e a execução contínua da aplicação mesmo em deslocamento.

Ambiente Ubíquo Disponibilizado

O ambiente ubíquo equivalente ao ambiente computacional onde recursos e serviços são gerenciados pelo EXEHDA com o propósito de atender os requisitos da computação ubíqua. A composição deste ambiente envolve tanto os dispositivos dos

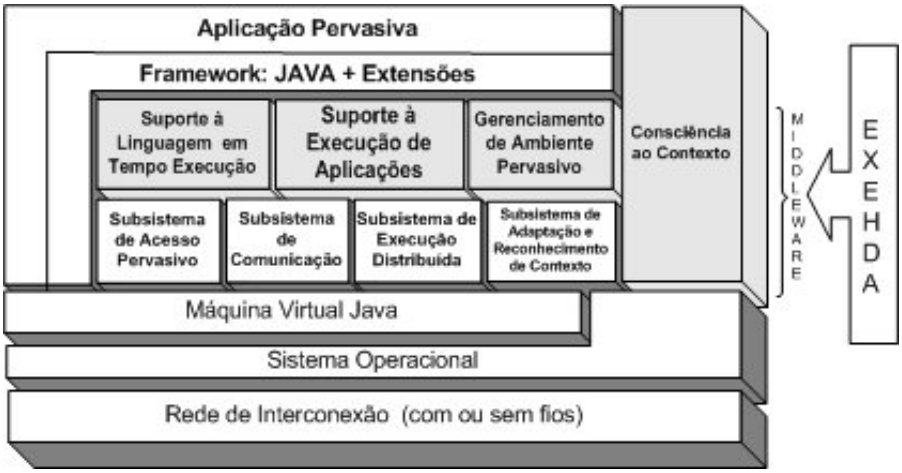

Figura 5.1: Arquitetura de Software do Middleware EXEHDA (YAMIN, 2004)

usuários, como os equipamentos da infraestrutura de suporte, todos instanciados pelo seu respectivo perfil de execução do *middleware*. A integração dos cenários da computação em grade, da computação móvel e da computação sensível ao contexto, é mapeada em uma organização composta pelo conjunto de células de execução do EXEHDA, conforme pode ser observado na figura 5.2 (ISAM, 2009).

O meio físico sobre o qual o ambiente ubíquo é definido constitui-se por uma rede infra estruturada, cuja composição final pode ser alterada pela agregação dinâmica de nodos móveis. Os recursos da infraestrutura física são mapeados para três abstrações básicas, as quais são empregadas na composição do ambiente ubíquo (YAMIN, 2004):

- EXEHDAcels: indica a área de atuação de uma EXEHDAbase, e é composta por esta e por EXEHDAnodos. Os principais aspectos considerados na definição da abrangência de uma célula são: o escopo institucional, a proximidade geográfica e o custo de comunicação;

- EXEHDAbase: é o ponto de convergência para os EXEHDAnodos. É responsável por todos os serviços básicos do ambiente ubíquo e, embora constitua uma referência lógica única, seus serviços, sobretudo por aspectos de escalabilidade, poderão estar distribuídos entre os vários dispositivos;

- EXEHDAnodo: são os dispositivos de processamento disponíveis no ambiente ubíquo, sendo responsáveis pela execução das aplicações. Um subcaso deste tipo de recurso é o EXEHDAnodo móvel. São os nodos do sistema com elevada portabilidade, tipicamente dotados de interface de rede para operação sem fio e, neste caso, integram a célula a qual seu ponto-de-acesso está subordinado. São funcionalmente análogos aos EXEHDAnodos, porém eventualmente com uma capacidade mais limitada (por exemplo, PDAs).

O ambiente ubíquo é formado por dispositivos multi-institucionais, o que gera a necessidade de adotar procedimentos de gerência iguais aos utilizados em ambientes de grade computacional (*Grid Computing*) (YAMIN, 2004). O gerenciamento da organização celular do ambiente ubíquo resguarda a autonomia das instituições envolvidas. Apesar de não contemplar mecanismos de gerência específicos para recursos especializados como impressoras, *scanners*, dentre outros o EXEHDA permite a catalogação de

tais recursos como integrantes de uma determinada célula do ambiente ubíquo, tornando-os, desta forma, passíveis de serem localizados dinamicamente e serem utilizados pelas aplicações ubíquas.

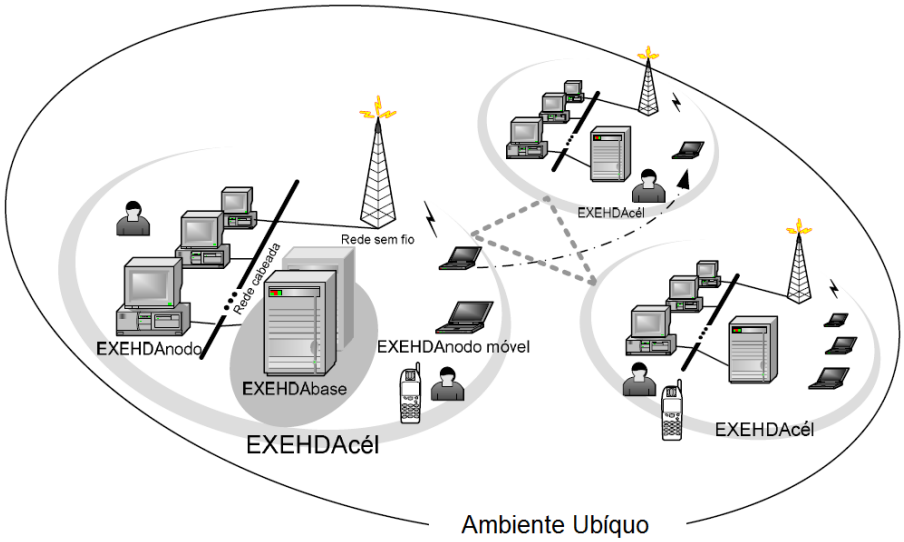

Figura 5.2: Ambiente Ubíquo Suprido pelo EXEHDA (YAMIN, 2004)

Composição de Serviços

A prerrogativa de operação em um ambiente altamente heterogêneo, onde não só o *hardware* exibe capacidades variadas de processamento e memória, mas também as bibliotecas de *software* disponíveis em cada dispositivo, motivaram a adoção de uma abordagem na qual um núcleo mínimo do *middleware* tem suas funcionalidades estendidas por serviços carregados sob demanda. Esta organização retrata um padrão de projeto referenciado na literatura como microkernel. Some-se a isto o fato de que esta carga sob demanda tem perfil adaptativo. Deste modo, poderá ser utilizada versão de um determinado serviço, melhor sintonizada às características do dispositivo em questão. Isto é possível porque, na modelagem do EXEHDA, os serviços estão definidos por sua interface, e não pela sua implementação propriamente dita.

A contraproposta à estratégia microkernel de um único binário monolítico, cujas funcionalidades cobrissem todas as combinações de necessidades das aplicações e dispositivos, se mostra inviável na computação ubíqua, cujo ambiente computacional apresenta uma alta heterogeneidade de recursos de processamento.

Entretanto, o requisito do *middleware* de manter-se operacional durante os períodos de desconexão planejada motivou, além da concepção de primitivas de comunicação adequadas a esta situação, a separação dos serviços que implementam operações de natureza distribuída em instâncias locais ao EXEHDAnodo (instância nodal), e instâncias locais a EXEHDAbase (instância celular). Neste sentido, o relacionamento entre instância de nodo e celular assemelha-se à estratégia de proxies, enquanto que o relacionamento entre instâncias celulares assume um caráter P2P (*Peer-to-Peer*). A abordagem P2P nas operações intercelulares vai ao encontro do requisito de escalabilidade. Uma organização dos subsistemas do EXEHDA pode ser observada na figura 5.3;

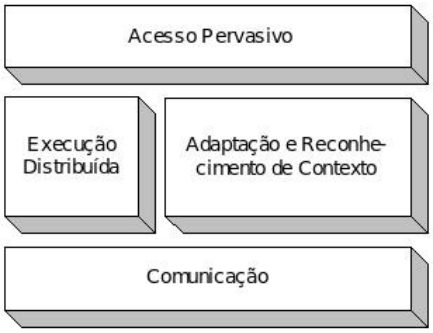

Figura 5.3: Organização dos Subsistemas do EXEHDA (YAMIN, 2004)

Sendo assim, os componentes da aplicação em execução em determinado dispositivo podem permanecer operacionais, desde que, para satisfação de uma dada requisição pelo *middleware*, o acesso a um recurso externo ao dispositivo seja prescindível. Por outro lado, a instância celular, em execução na base da célula, provê uma referência para os outros recursos, no caso da realização de operações que requeiram coordenação distribuída. Portanto, observe-se que a EXEHDAbase é, por definição, uma entidade estável dentro da EXEHDAcel, permitindo que os demais integrantes (recursos) da célula tenham um caráter mais dinâmico com relação a sua disponibilidade (presença estável) na célula.

O Núcleo do EXEHDA

A funcionalidade servida pelo EXEHDA é personalizável no nível de nodo, sendo determinada pelo conjunto de serviços ativos e controlada por meio de perfis de execução. Um perfil de execução define um conjunto de serviços a ser ativado em um EXEHDAnodo, associando a cada serviço uma implementação específica dentre as disponíveis, bem como definindo parâmetros para sua execução. Adicionalmente, o perfil de execução também controla a política de carga a ser utilizada para um determinado serviço, a qual se traduz em duas opções: (i) quando da ativação do nodo (*bootstrap* do *middleware*) e (ii) sob demanda.

Deste modo, a informação definida nos perfis de execução é também consultada quando da carga de serviços sob demanda, assim, a estratégia adaptativa para carga dos serviços acontece tanto na inicialização do nodo, quanto após este já estar em operação e precisar instalar um novo serviço. Esta política para carga dos serviços é disponibilizada por um núcleo mínimo do EXEHDA, o qual é instalado em todo EXEHDAnodo que for integrado ao ambiente ubíquo. Este núcleo é formado por dois componentes, conforme observado na figura 5.4:

- ProfileManager: interpreta a informação disponível nos perfis de execução e a disponibiliza aos outros serviços do *middleware*. Cada EXEHDAnodo tem um perfil de execução individual;
- ServiceManager: Faz a ativação dos serviços no EXEHDAnodo a partir das informações disponibilizadas pelo ProfileManager. Para isto, carrega sob demanda o código dos serviços do *middleware*, a partir do repositório de serviços que pode ser local ou remoto, dependendo da capacidade de armazenamento do EXEHDAnodo e da natureza do serviço.

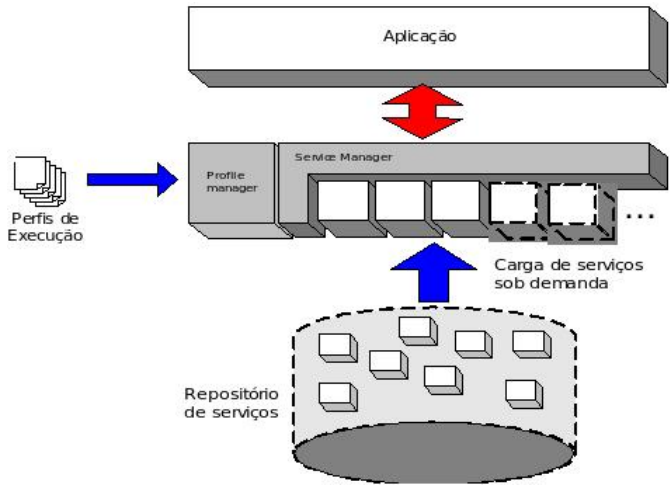

Figura 5.4: Organização do Núcleo do EXEHDA (YAMIN, 2004)

5.2.1.3 Subsistemas do EXEHDA

O *middleware* EXEHDA é formado por quatro subsistemas: Subsistema de Execução Distribuída, Subsistema de Comunicação, Subsistema de Acesso Ubíquo e Subsistema de Reconhecimento de Contexto e Adaptação. A seguir estão resumidas as funcionalidades de cada um desses subsistemas.

Subsistema de Execução Distribuída

O suporte ao processamento distribuído no EXEHDA e de responsabilidade do subsistema de execução distribuída. Com a finalidade de promover uma execução efetivamente ubíqua, este subsistema interage com outros subsistemas do EXEHDA. Em específico, interage com o subsistema de reconhecimento de contexto e adaptação, de forma a prover comportamento distribuído e adaptativo às aplicações da computação ubíqua. Este subsistema é formado pelos serviços: (i) *Executor*, (ii) *Cell Information Base* - CIB, (iii) *OXManager*, (iv) *Discoverer*, (v) *ResourceBroker*, (vi) *Gateway*, (vii) *StdStreams*, (viii) *Logger*, e (ix) *Dynamic Configurator* - DC.

Subsistema de Comunicação

Devido a mobilidade do *hardware*, assim como a do *software*, não garante a interação contínua entre os componentes da aplicação distribuída. As desconexões são frequentes, não somente devido à existência de alguns links sem fio, mas sobretudo como uma estratégia para economia de energia nos dispositivos móveis. O subsistema de comunicação do EXEHDA disponibiliza mecanismos que atendem estes aspectos da computação ubíqua. Fazem parte deste subsistema os serviços *Dispatcher*, *WORB*, *CC-Manager* os quais contemplam modelos com níveis diferenciados de abstração para as comunicações.

Subsistema de Acesso Ubíquo

Devido a premissa de acesso em qualquer lugar, todo o tempo, a dados e código da computação ubíqua, requer um suporte do *middleware*. Os serviços que integram este

subsistema no EXEHDA são: (i) BDA - Base de Dados Ubíqua das Aplicações, (ii) AVU - Ambiente Virtual do Usuário, (iii) *SessionManager* e (iv) *Gatekeeper*.

Subsistema de Reconhecimento de Contexto e Adaptação

Compõem este subsistema os serviços (i) *Collector*, (ii) *Deflector*, (iii) *ContextManager*, (iv) *AdaptEngine* e (v) *Scheduler*. Particularmente, *AdaptEngine* e *Scheduler* são responsáveis, respectivamente, pelo controle das adaptações de cunho funcional e não-funcional. Compreende-se por adaptação funcional aquela que implica a modificação do código sendo executado. Já a adaptação não-funcional, é aquela que atua sobre a gerência da execução distribuída, na qual podem ser identificados seus diversos serviços (YAMIN, 2004).

5.3 O Serviço EXEHDA-SS

O EXEHDA-SS (VENECIAN, 2010) constitui uma recente contribuição ao Subsistema de Reconhecimento de Contexto e Adaptação, tendo sido concebido para qualificar o mecanismo de sensibilidade ao contexto. O mesmo será parte importante na arquitetura de software do uMED.

Aplicações ubíquas executam em ambientes instrumentados com sensores, geralmente dotados de interfaces de redes sem fio, nos quais dispositivos, agentes de *software* e serviços são integrados e cooperam para atender os objetivos dos usuários finais. Essa categoria de aplicações caracteriza-se por constantes mudanças em seu contexto de execução, geradas pelos ambientes altamente dinâmicos em que tipicamente executam. Nesta perspectiva introduzida pela computação ubíqua (COSTA; YAMIN; GEYER, 2008), o processamento está espalhado no ambiente através de vários dispositivos, que executam tarefas bem definidas dependendo de sua natureza, interligados de forma que a infraestrutura de comunicações torne-se o mais imperceptível possível para o usuário.

Deste modo, a sensibilidade ao contexto refere-se à capacidade de um sistema computacional perceber características de seu ambiente, e é um requisito fundamental para permitir a adaptação em resposta às mudanças ambientais. Aplicações sensíveis ao contexto conhecem o ambiente no qual estão sendo utilizadas e tomam decisões de acordo com mudanças no seu próprio ambiente. Ou seja, reagem a ações executadas por outras entidades, podendo essas ser pessoas, objetos ou até mesmo outros sistemas, que modifiquem o ambiente. Essas aplicações, de um modo geral, tomam ciência das alterações que venham a acontecer no ambiente. Tais modificações denominam-se alterações nas informações de contexto (VENECIAN, 2010).

Segundo Venecian (VENECIAN, 2010) um ambiente ubíquo tem uma natureza dinâmica, devido à mobilidade, a diversidade de dispositivos e tecnologias existentes, assim como às mudanças constantes no ambiente computacional. Ao se construir e executar aplicações ubíquas sensíveis ao contexto, há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir de um conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, e a realização de inferências para seu uso em tomadas de decisão .

O mecanismo concebido utiliza o ambiente ubíquo promovido pelo *middleware* EXEHDA (YAMIN, 2004). Este se propõem a integração de tecnologias de suporte semântico em mecanismo de sensibilidade ao contexto, desde a sua aquisição, processamento e distribuição das informações contextuais, direcionado a computação ubíqua. E

denomina-se EXEHDA-SS (Suporte Semântico) (VENECIAN, 2010).

O EXEHDA-SS (VENECIAN, 2010) foi concebido para ser responsável pelo tratamento das informações contextuais, realizando tarefas de manipulação e dedução sobre o contexto, utilizando ontologias para suporte a representação e processamento das informações contextuais. Através do uso de inferências será possível um refinamento qualificado dessas informações capturas e distribuídas nas células de execução do EXEHDA.

5.3.1 Concepção e Modelagem do EXEHDA-SS

Na continuidade será tratado o modelo de representação de contexto do ambiente ubíquo e a arquitetura de *software* proposta para sensibilidade ao contexto, personalizável por componentes de *software* das aplicações com suporte a processamento semântico do EXEHDA-SS.

Modelo de Representação de Contexto

Na concepção do EXEHDA-SS, foi preferido o uso de ontologias como mecanismo para representação e processamento de contexto (VENECIAN, 2010). Ontologias têm sido largamente utilizadas em áreas como gerenciamento de conteúdo, conhecimento e comércio eletrônico. De um modo geral, ontologias têm sido usadas para representar ambientes ubíquos, descrevendo, comumente, entidades envolvidas e suas respectivas propriedades. Elas definem principalmente os diferentes tipos de aplicações, serviços, dispositivos, sensores, entre outros.

Conforme Venecian (VENECIAN, 2010) as ontologias no EXEHDA-SS são empregadas sob duas perspectivas: (i) representação semântica dos dados contextuais e (ii) o estabelecimento de relações entre os mesmos, possibilitando a realização de inferências. Através do uso de inferências espera-se contribuir para qualificação das informações contextuais a serem entregues aos demais serviços do EXEHDA e/ou aplicações que tenham interesse em manipulação de dados contextuais.

Como núcleo do modelo semântico para representação de contexto é utilizado, a OntContext, ontologia responsável pela representação dos contextos coletados, notificados e das instâncias dos contextos de interesse das aplicações. A OntContext, figura 5.5, é empregada pelo Servidor de Contexto do EXEHDA-SS para prover suporte semântico ontológico nas tarefas de dedução utilizando informações contextuais.

Na estrututa de classes da OntContext, foi definida a classe Sensor, que contém as instâncias dos sensores que participam do sensoramento do ambiente ubíquo provido pelo *middleware*. A seguir são detalhadas as demais classes e subclasses que compõem a OntContext.

Classes, Subclasses e Atributos para Coleta e Notificação de Contextos

- Contexto: A classe contexto responsável por armazenar os dados coletados pelo sensores como: identificador do nodo, data/hora, usuário, identificador do contexto de interesse, sensor, valor coletado pelo sensor e valor que ocorreu tradução.

- Contexto_Notificado: Esta classe contém os contextos notificados com os seguintes atributos: identificador, aplicação, componente, adaptador e usuário.

- ContextoNotificado_Sensor: Esta subclasse possui identificação e o sensor que gerou uma notificação, além do valor coletado pelo respectivo sensor, identificador do nodo e valor traduzido.

- ContextoNotificado_Deduzido: São armazenados nesta subclasse os identificadores das regras de dedução e valores deduzidos.

A classe Contexto_Notificado e as subclasses ContextoNotificado_Sensor e ContextoNotificado_Deduzido armazenam as informações contextuais processadas e deduzidas.

Classes e Atributos do Contexto de Interesse das Aplicações

- Contexto_Interesse: Contém o identificador, aplicação, componente e adaptador.
- Sensor_Public: Define os parâmetros operacionais dos sensores, como: intervalo do tempo de medicação, taxa de flutuação mínima dos dados a serem publicados, valor inferior e superior, valor *default* do sensor, regra para tradução de dados coletados e número máximo de publicações.
- Contexto_Deduzido: Fazem parte desta classe os atributos identificador e a regra de dedução. Essas regras são processadas pelo Motor de Inferência do Gerente de Interpretação do EXEHDA-SS.

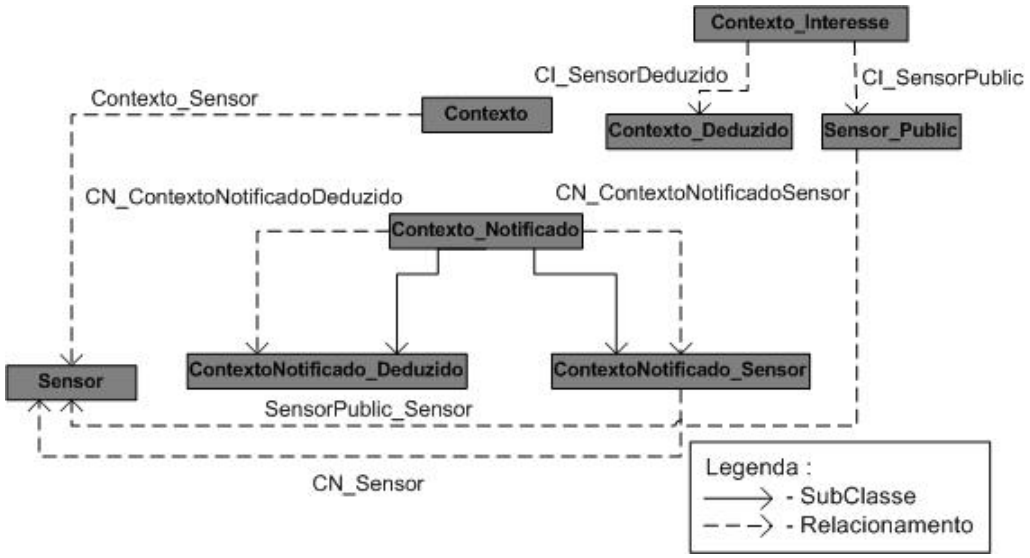

Figura 5.5: Classes da OntContext (VENECIAN, 2010)

5.3.2 Arquitetura de Software do EXEHDA-SS

Segundo Venecian (VENECIAN, 2010) a arquitetura de *software* do EXEHDA-SS é direcionada ao processamento das informações contextuais fazendo o uso de tecnologias de suporte semântico nas funcionalidades de aquisição, processamento e distribuição das informações de contexto processadas.

Desta forma, o EXEHDA-SS é alimentado por contextos de interesses das aplicações. Esses contextos de interesses são responsáveis por caracterizar os aspectos que devem ser considerados nos procedimentos de monitoração do ambiente ubíquo, de interpretação destes dados capturados e das respectivas notificações.

O Servidor de Contexto é composto por três serviços: O Gerente de Aquisição, Gerente de Interpretação e Gerente de Notificação. Esses gerentes são autônomos e cooperantes para a realização de tarefas de manipulação e dedução sobre o contexto.

Assim, o EXEHDA-SS mostra-se alinhado para prover suporte semântico na tarefas de aquisição e notificação das informações contextuais aos demais serviços do EXEHDA e/ou as aplicações. O emprego de ontologias se mostra oportuno para prover o suporte semântico necessário.

5.4 Considerações sobre o Capítulo

Este capítulo apresentou uma revisão dos fundamentos utilizados pelo uMED como o projeto PERTMED, o *middleware* EXEHDA, abordando suas premissas de pesquisa, sua organização e subsistemas. Também foi analisado o EXEHDA-SS um mecanismo de sensibilidade ao contexto empregado neste *middleware*, abordando os fundamentos, concepção e modelagem de *software*. O próximo capítulo apresenta uma arquitetura de *software* direcionada a área médica.

6 UMED: CONCEPÇÃO E MODELAGEM

Este capítulo resume as decisões de concepção e modelagem do uMED. O *framework* uMED está inserido nos esforços de pesquisa do projeto PERTMED (Sistema de TeleMedicina Móvel) (PERTMED, 2007), no qual vem sendo desenvolvido por três universidades do sul do Brasil (Universidade Federal de Santa Maria, Universidades Católica e Federal de Pelotas) contando com a participação dos grupos de pesquisa (GMob e G3PD), e com a colaboração de equipes médicas ligadas a essas universidades. O uMED é uma arquitetura de *software* direcionada para promover o desenvolvimento de aplicações para medicina ubíqua.

6.1 Arquitetura de Software do uMED

O *framework* uMED tem como premissa tratar as informações adquiridas pelos sensores, notificar as partes interessadas conforme regras de tratamento de contexto utilizadas pelas diferentes aplicações e possibilitar que os sensores e/ou os atuadores possam ser ativados, desativados, configurados e consultados a qualquer hora de forma ubíqua, além de permitir alterações de regras de tratamento de contexto mesmo durante a execução das aplicações.

Deste modo, o uMED direciona-se para aplicações da área médica nas quais é necessário monitorar pacientes por sensores, e gerar de forma autônoma diferentes níveis de alerta conforme regras definidas pelos profissionais de saúde.

Uma visão da arquitetura de *software* do uMED é apresentado na figura 6.1. A mesma é composta por cinco módulos: Gerente de Atuação, Gerente de Aplicações, Gerente de Borda, Gerente de Comunicação e o Servidor de Contexto. Os serviços da arquitetura de *software* são autônomos e cooperantes e possibilitam coleta de dados, assim como controle da ativação, da desativação e configuração de atuadores. A arquitetura conta com diversos repositórios de dados, os quais são atualizados e/ou consultados durante a operação da mesma.

O *framework* uMED, emprega o ambiente ubíquo definido no projeto ISAM (ISAM, 2009) que é provido pelo *middleware* EXEHDA (YAMIN, 2004).

O uMED irá contemplar o uso de tecnologias de processamento semântico quando da execução das regras de tratamento de contexto.

Para que a arquitetura de *software* proposta, para o uMED, possa realizar tarefas de dedução sobre o contexto, a mesma é alimentada por contextos de interesse das diversas aplicações.

No Contexto de Interesse das Aplicações ficam definidas quais informações con-

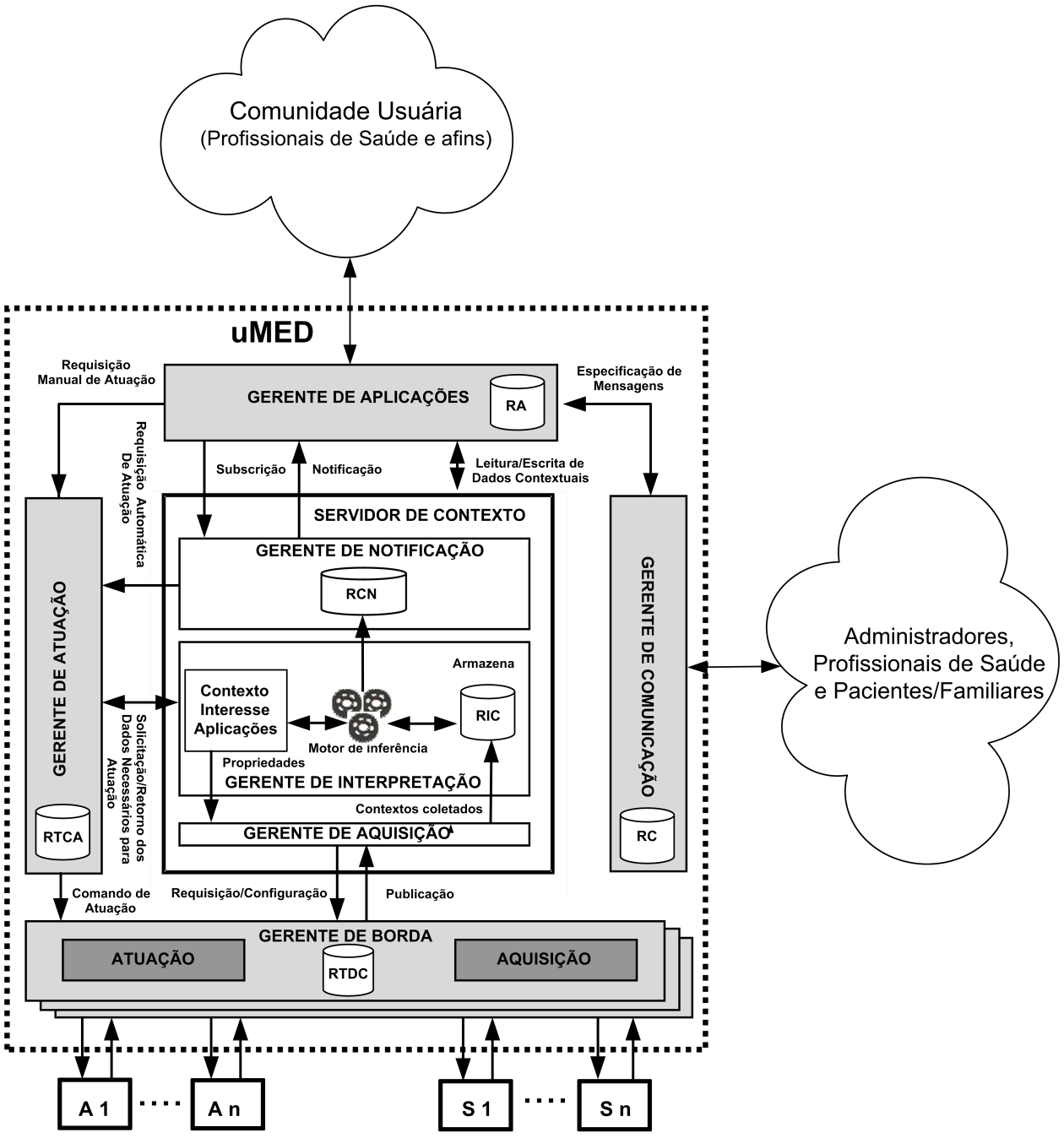

Figura 6.1: Arquitetura de Software Proposta para uMED

textuais serão adquiridas pelos sensores, e processadas (traduzidas e deduzidas) pela arquitetura, bem como quais comandos serão empregados no controle dos atuadores. Essas especificações são relacionadas por aplicação, componente e adaptador, considerando os seguintes parâmetros e regras:

- instanciação dos atuadores e sensores que participam das aplicações;
- parâmetros operacionais para controle da ativação, da desativação, bem como da configuração dos sensores e/ou atuadores;
- especificação do número máximo de registros pelos sensores a serem armazenados no repositório contextual. Ultrapassando este número, os valores mais antigos coletados pelos sensores serão apagados;
- regras de deduções sobre os dados adquiridos pelos sensores;
- regras de deduções sobre os dados utilizados para controle dos atuadores (dados de ativação).

uMED: Principais Módulos

As próximas seções discutem a concepção dos módulos que integram o uMED. Neste sentido, são exploradas as decisões de projeto e as funcionalidades de cada um dos seus módulos: Gerente de Atuação, Gerente de Aplicações, Gerente de Borda e o Gerente de Comunicação, além desses gerentes também compõem a arquitetura de *software* do uMED um Servidor de Contexto.

6.1.1 Gerente de Atuação

Responsável pelo controle da ativação, da desativação e configuração dos atuadores empregados pelas aplicações gerenciadas pelo *framework* uMED, este Gerente possui as seguintes funcionalidades:

- processar os contextos de interesse da aplicação, extraindo as informações necessárias para sua operação;
- coordenar juntamente com o Gerente de Borda a ativação, a desativação e a configuração dos parâmetros operacionais dos atuadores;
- notificar ao Gerente de Borda quais atuadores serão necessários para atender as demandas das aplicações em uso;
- disparar no ambiente ubíquo rotinas de verificação do status dos diferentes atuadores considerando os seus aspectos operacionais.

O Gerente de Atuação trabalha com duas abordagens distintas: (i) Atuação Manual; (ii) Atuação Automática. A seguir são discutidas estas duas abordagens.

Atuação Manual

Compreende-se como uma atuação manual aquela que é disparada pelo usuário da aplicação a qualquer instante, ou seja, um profissional de saúde deseja ativar um dispositivo qualquer, ou alterar algum parâmetro operacional do mesmo. Os dispositivos podem ser de qualquer natureza, por exemplo, aquecedores, umidificadores, alarmes, luzes de sinalização, e até mesmo dipositivos com relação com demandas médicas: aparelho de oxigênio, bomba de infusão, ventilador pulmonar, entre outros.

Atuação Automática

A atuação automática é disparada através notificações produzidas pelo Servidor de Contexto, ou seja, o processamento de um determinado contexto deduzido poderá implicar a ativação e/ou desativação e/ou configuração de um dispositivo. Os parâmetros operacionais necessário para atuação são obtidas do próprio Servidor de Contexto (vide tabela 6.1).

Para tratamento da perda de conexão entre o Gerente de Atuação e o Gerente de Borda, existe mecanismo de bufferização dos comandos emitidos, o qual se vale de um parâmetro do tipo *time-to-live* para avaliar quais dos comandos ainda tem sentido ser tratado pelo Gerente de Borda.

Tabela 6.1: Parâmetros Operacionais dos Atuadores

Parâmetros dos Atuadores	**Objetivo**
Identificador da Aplicação	Definir um parâmetro único de identificação da aplicação
Identificador do Componente	Definir um parâmetro único de identificação para cada componente das aplicações
Identificador do Adaptador	Definir um parâmetro único de identificação para cada adaptador associado a um componente
Identificador do Atuador	Definir um parâmetro único de identificação de um atuador
Status do Atuador	Registrar o estado dos atuadores
Potência de Atuação	Definir a potência de operação para os atuadores (se for o caso)
Intervalo de Atuação	Especificar o intervalo em que devem permanecer operacionais os atuadores (a unidade é segundo)

Repositório do Gerente de Atuação

O Gerente de Atuação contempla um repositório denominado RTCA (Repositório Temporário de Comandos de Atuação), objetivo deste repositório é atuar registrando (bufferizando) os comandos de atuação que eventualmente não puderem ser transmitidos ao Gerente de Borda, quando de uma interrupção temporária nas comunicações.

Fluxos de Informações do Gerente de Atuação

Nesta seção são discutidos os fluxos pertinentes ao Gerente de Atuação (vide figura 6.1).

O fluxo de informação entre Gerente de Aplicações e o Gerente de Atuação é denominado “Requisição Manual de Atuação” neste fluxo trafegam informações produzidas pelo Gerente de Aplicações para ativação, desativação ou configuração de algum atuador, isto concede ao usuário da aplicação a possibilidade de intervir no meio físico manualmente - “Atuação Manual”.

Existem dois fluxos entre o Servidor de Contexto e o Gerente de Atuação: (i) o primeiro denominado de “Requisição Automática de Atuação” trafegam informações produzidas pelo Servidor de Contexto para ativação e/ou desativação de um atuador sem a intervenção do usuário - “Atuação Automática”; (ii) o segundo cuja denominação é “Solicitação/Retorno dos Dados Necessários para Atuação” (fluxo bidirecional) trafegam os parâmetros operacionais necessários para atuação. O Gerente de Atuação solicita estes parâmetros ao Servidor de Contexto o qual envia os dados requisitados. Este fluxo se faz necessário para que o Gerente de Atuação receba os parâmetros operacionais necessários para interagir com os atuadores que fazem parte de um determinado contexto, como por exemplo, acionar uma campainha na potência máxima por um período de tempo de 60 segundos e acender um sinal luminoso na potência máxima por um período de tempo de 120 segundos em uma determinada enfermaria do hospital.

O fluxo de informação entre o Gerente de Atuação e o Gerente de Borda é denominado de “Comando de Atuação” este fluxo é formado por vários parâmetros operacionais destinados aos atuadores. A relação entre os mesmos pode ser observada na figura 6.2.

6.1.2 Gerente de Aplicações

Nesta seção é apresentado o Gerente de Aplicações o qual é responsável por disponibilizar ao usuário final as aplicações que estão sendo gerenciadas pelo *framework* uMED. O Gerente de Aplicações trabalha com duas abordagens distintas: (i) componentes de *software* das aplicações assíncronos ao contexto; (ii) componentes de *software* das aplicações síncronos ao contexto, ambas possuem características complementares quando da definição das aplicações ubíquas do uMED.

Para este trabalho compreende-se como um componente de *software* de aplicação síncrono ao contexto aquele cuja execução é dispara pelo uMED caso ocorra algum evento de interesse do contexto da aplicação, já por componente de *software* de aplicação assíncrono ao contexto temos aqueles cuja execução ocorre por intervenção do usuário. A seguir é detalhado a utilização destas duas abordagens empregadas pelo Gerente de Aplicações.

Componentes de Software das Aplicações Síncronos ao Contexto

Uma das atribuições do Gerente de Aplicações é disponibilizar componentes de *software* das aplicações que são síncronos ao contexto, via de regra, estes componentes são destinados ao monitoramento de sinais vitais de pacientes ou do ambiente relacionado aos mesmos, e emitir de forma automatizada, diferentes níveis de alertas, em função dos dados coletados, para os profissionais de saúde (médicos, enfermeiros, entre outros).

As aplicações do uMED se subscrevem ao Servidor de Contexto e aguardam as notificações geradas por este servidor. Por exemplo, uma aplicação se subscreve informando se a temperatura corporal ou a pressão arterial de um determinado paciente alterar 10%, automaticamente deseja-se que essa variação seja notificada a um determinado profissional de saúde, essa notificação pode ser através de envio de mensagens ou pode ser

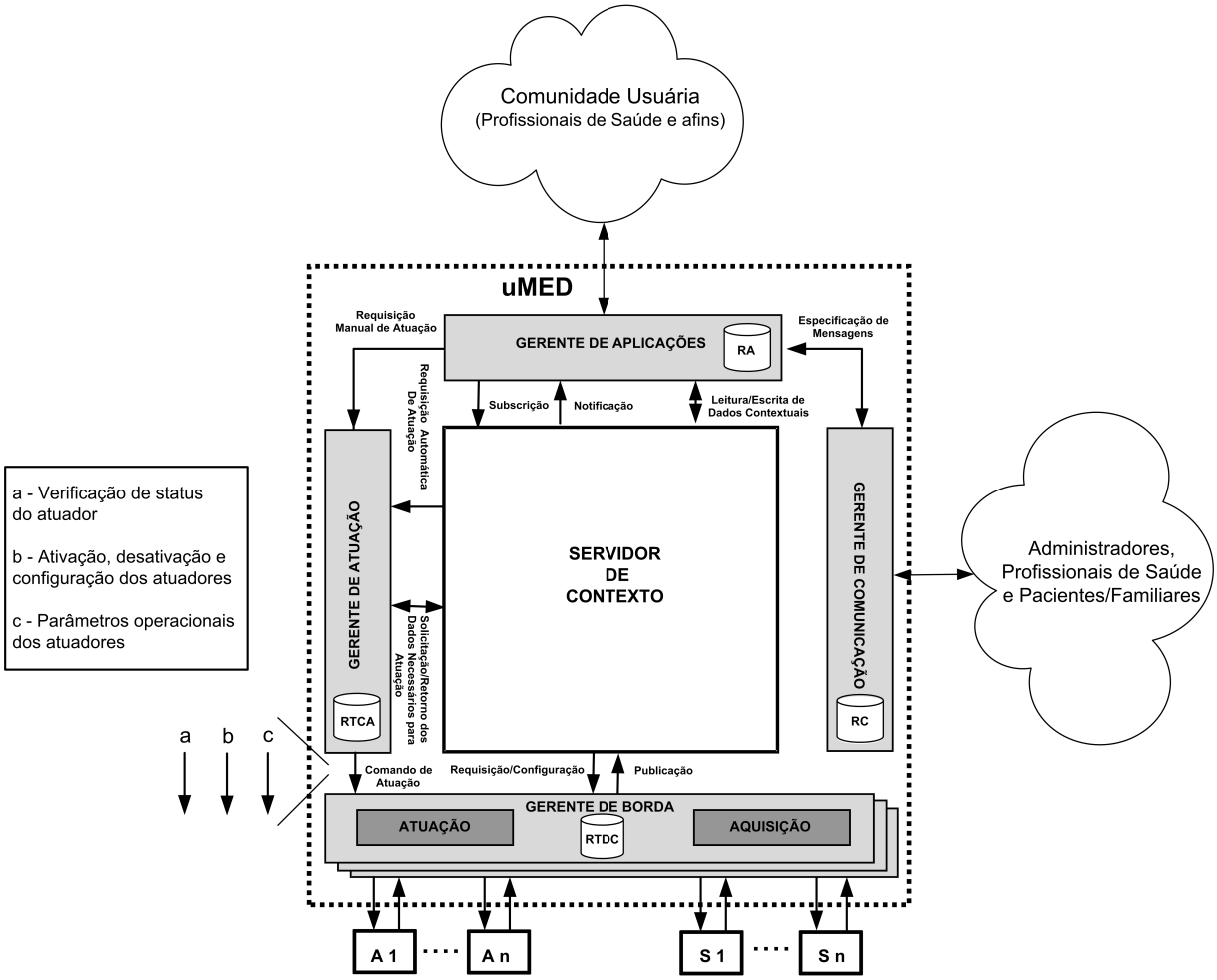

Figura 6.2: Fluxo de Informações para o Gerente de Borda

um disparo de um alerta em uma determinada enfermaria, ou ambas as notificações. É prevista a possibilidade de alteração das regras em tempo de execução, ou seja, uma determinada regra anteriormente submetida ao Servidor de Contexto pode sofrer alterações a qualquer instante.

Componentes de Software das Aplicações Assíncronos ao Contexto

A outra atribuições do Gerente de Aplicações é disponibilizar os componentes de *software* das aplicações que são assíncronos ao contexto, provendo as seguintes funcionalidades: (i) permitir a manipulação instantânea de sensores e atuadores; (ii) atuar no ambiente ubíquo através de alterações nas configurações operacionais dos atuadores; (iii) conceder ao usuário da aplicação a possibilidade de criar relatórios personalizados.

Os componentes de *software* das aplicações assíncronos ao contexto, por sua vez, permitem que os profissionais de saúde (médicos, enfermeiros, entre outros) possam consultar informações sobre os dados de um determinado paciente a qualquer hora, em qualquer lugar (acesso ubíquo), e disponibilizar utilizando os mesmos séries históricas para auxiliar no diagnóstico.

Relatórios Personalizados

Está previsto para o Gerente de Aplicações a funcionalidade de emitir relatórios personalizáveis pelos usuários do sistema. Estão previstos dois tipos de relatórios: de MONITORAMENTO e ATUAÇÃO. No relatório de MONITORAMENTO será facultado ao usuário da aplicação obter uma listagem dos dados coletados, por sua vez, no relatório de ATUAÇÃO será concedido ao usuário da aplicação uma listagem referente aos dados que foram utilizados pelos atuadores. Estes tipos de relatórios podem ser personalizados através de duas formas gerais: (i) por período de tempo; (ii) através de faixa de valores dos dipositivos.

Também é possível criar relatórios a respeito das mensagens enviadas denominado relatório de STATUS DE MENSAGEM ENVIADAS, neste relatório seria possível visualizar todas as mensagens transmitidas que obteram êxito em sua entrega "Status Sucesso" ou as que falharam em sua transmissão "Status Pendente". Para este relatório também é possível personalizar através de duas formas gerais: (i) por período de tempo; (ii) através do profissional de saúde notificado.

Este relatório do tipo STATUS DE MENSAGEM ENVIADAS não irá tratar do status das mensagens instantâneas através da rede celular (mensagens síncronas) conforme será abordado na seção 6.1.4.

Um outro relatório possível seria o relatório de HISTÓRICO CLÍNICO, permitindo assim aos profissionais de saúde uma análise mais aprofundada do caso clínico do paciente, pois neste relatório estariam registradas todas as informações coletadas pelos sensores, assim como, todos as ações dos atuadores utilizados em um paciente, em um determinado período de tempo.

Estes relatórios de históricos poderiam ser utilizados como material didático destinado a formação dos profissionais de saúde, ou até mesmos, para discussão de casos clínicos pelas juntas médicas, possibilitando assim segunda opinião de outros profissionais de saúde.

Repositório do Gerente de Aplicações

O Gerente de Aplicações contempla um repositório denominado RA (Repositório de Aplicações), o qual contém os componentes de *software* que compõem as aplicações a serem executadas, bem como os parâmetros operacionais destas aplicações administradas. Este repositório funciona como um *container* Web disponibilizando as funcionalidades providas pelas diferentes aplicações.

Fluxos de Informações do Gerente de Aplicações

Neste seção é apresentado os fluxos pertinentes ao Gerente de Aplicações (vide figura 6.1).

Toda vez que uma aplicação é iniciada o Gerente de Aplicações subscreve os aspectos pertinentes a mesma no Servidor de Contexto através de um fluxo cuja denominação é "Subscrição". Neste fluxo, dentre as informações que trafegam pode ser destacado o Contexto de Interesse das Aplicações. Este mesmo Gerente aguarda ser notificado através de um fluxo denominado "Notificação" através do qual recebe notificações produzidas pelo Servidor de Contexto, a partir das quais são disparados os componentes de *software* pertinentes.

Por sua vez, o fluxo de informações entre o Gerente de Aplicações e o Servidor de Contexto é denominado de "Leitura/Escrita de Dados Contextuais" (fluxo bidirecional) este fluxo se faz necessário para a leitura ou alteração de algum dado contextual.

O fluxo de informações entre o Gerente de Aplicações e o Gerente de Comunicação, cuja denominação é "Especificação de Mensagens" (fluxo bidirecional), tem por objetivo trafegar parâmetros operacionais tais como: Id do contexto de interesse, tipo de mensagem entre outras. Estes parâmetros, juntamente com informações de contato disponíveis no Repositório de Comunicação, como por exemplo, número do telefone celular, email, conta do Gtalk são indispensáveis ao Gerente de Comunicação para a administração de mensagens de alerta para pessoas envolvidas em um determinado contexto.

Neste mesmo fluxo ("Especificação de Mensagens") também é possível que o Gerente de Aplicações solicite dados existentes no Repositório de Comunicação para que seja possível criar relatórios do tipo STATUS DE MENSAGEM ENVIADAS.

Por sua vez, o fluxo de informações entre o Gerente de Aplicações e o Gerente de Atuação que é denominado "Requisição Manual de Atuação" foi discutido na seção 6.1.1.

6.1.3 Gerente de Borda

Este Gerente é responsável pela aquisição de dados capturados pelos sensores, assim como, pelo controle dos atuadores. Os dados adquiridos pelos sensores passam por várias etapas, nas quais é feito o tratamento apropriado, de modo que seja possível então a persistência dos mesmos. Este dado persistido, é publicado no Servidor de Contexto para ser analisado.

A arquitetura contempla que os contextos possam ser capturados por diferentes Gerentes de Borda, cada um com conjunto particular de sensores e/ou atuadores com regimes operacionais distintos. Estes Gerentes de Borda podem estar distantes entre si bem como dos outros módulos do uMED.

O controle dos atuadores depende dos parâmetros operacionais feitos pelo Gerente de Atuação o qual considera as preferências do Contexto de Interesse das Aplicações.

No Gerente de Borda, sensores e/ou atuadores podem ser implantados por meio de arquivos de configuração. Desta forma, um sensor e/ou atuador pode ser adicionado, removido ou substituído por outra implementação mais adequada ao dispositivo onde está sendo empregado no uMED.

Para a aquisição dos contextos através de sensores se faz necessário:

- especificar intervalos de tempo entre medições;
- registrar flutuação mínima para que aconteça a publicação;
- definir a faixa na qual os valores dos sensores deverão ser publicados.

O Servidor de Contexto envia parâmetros operacionais ao Gerente de Borda através de um arquivo XML (*eXtensible Markup Language*), o que contribui para modularidade da arquitetura. Conforme pode ser observado na figura 6.3 o arquivo chamado de `SensorConfiguracao`, contém parâmetros para inicialização e publicação de contextos pelos sensores ao Servidor de Contexto as informações contextuais utilizada pelos sensores podem ser observadas na tabela 6.2.

Tabela 6.2: Parâmetros Operacionais dos Sensores

Parâmetros dos Sensores	**Objetivo**
Identificador da Aplicação	Definir um parâmetro único de identificação da aplicação
Identificador do Componente	Definir um parâmetro único de identificação para cada componente das aplicações
Identificador do Adaptador	Definir um parâmetro único de identificação para cada adaptador associado a um componente
Identificador do Sensor	Definir um parâmetro único de identificação de um sensor
Frequência de Publicação	Especificar o intervalo em que deve ser feita a leitura dos sensores (a unidade é segundos)
Flutuação Mínima	Especificar o valor mínimo que deve variar o valor coletado para que ocorra uma publicação
Valor Mínimo	Especificar o valor mínimo para que ocorra publicação de um sensor
Valor Máximo	Especificar o valor máximo para que ocorra publicação de um sensor

O arquivo de configuração deverá ser utilizado para definir os parâmetros operacionais dos sensores de acordo com as especificações necessárias para a aplicação e registradas no Contexto de Interesse das Aplicações. Como exemplo destes parâmetros têm-se: identificador do sensor, frequência de publicação, flutuação mínima, faixa mínima e máxima.

```
<?xml version="1.0" enconding="UTF-8"?>
<Sensors>
<Sensor identificador="codigo_sensor1" intMed="frequencia_publicacao"
flut="flutuacao_minima" valInf="faixa_minima" ValSup="faixa_maxima"/>
<Sensor identificador="codigo_sensor2" intMed="frequencia_publicacao"
flut="flutuacao_minima" valInf="faixa_minima" valSup="faixa_maxima"/>
</Sensors>
```

Figura 6.3: Exemplo de Arquivo `SensorConfiguracao`

De forma análoga, o atuador também deve ser configurado por um arquivo XML. O Gerente de Atuação envia parâmetros operacionais para o Gerente de Borda, em um arquivo XML o qual contém os parâmetros operacionais definidos no Contexto de Interesse das Aplicações. Esses parâmetros especificam parâmetros necessários para atuação, tais como: identificador do atuador, ativa/desativa atuador, potência de atuação, intervalo de atuação. O arquivo `AtuadorConfiguracao` pode ser observado na figura 6.4, que registra os mesmos.

O Gerente de Borda é composto de dois grandes módulos: módulo Aquisição e módulo Atuação. Por sua vez, cada um desses módulos compreendem partes menores promovendo uma divisão da arquitetura em partes com elevada coesão funcional (vide figura 6.5).

```
<?xml version="1.0" enconding="UTF-8"?>
<Atuadors>
<Atuador identificador="codigo_atuador1" intMed="Status_atuador"
pot="potencia_atuacao" intAtua="tempo_segundos"/>
<Atuador identificador="codigo_atuador2" intMed="Status_atuador"
pot="potencia_atuacao" intAtua="tempo_segundos"/>
</Atuadors>
```

Figura 6.4: Exemplo de Arquivo `AtuadorConfiguracao`

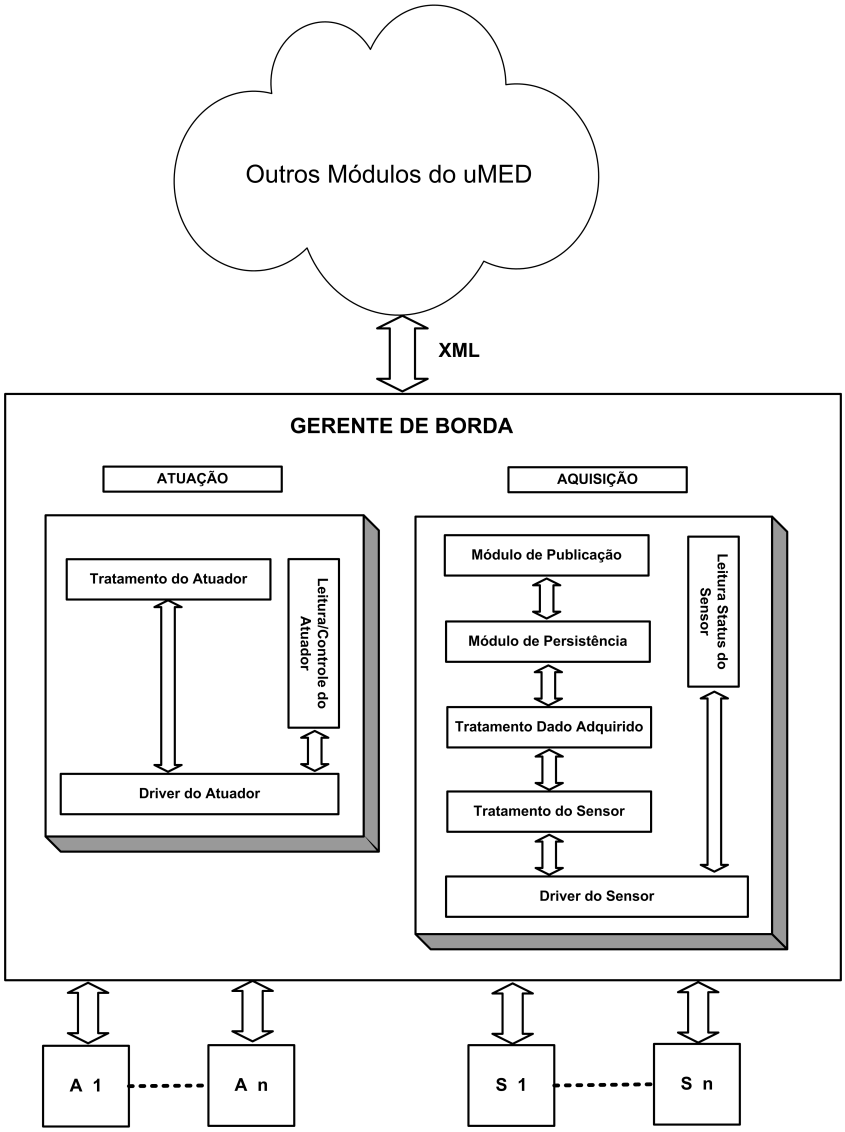

Figura 6.5: Vista Expandida do Gerente de Borda

Uma descrição do módulo Aquisição e do módulo Atuação é feita a seguir.

1. Módulo Aquisição

   O módulo Aquisição do Gerente de Borda possui várias partes cujas funcionalidades estão caracterizadas a seguir:

   - Driver do Sensor - encapsula o tratamento de baixo nível dos sensores de modo a facilitar a inserção de novos sensores no Gerente de Borda, este driver implementa uma interface padrão. Desta forma, a rotina de leitura dos sensores poderá ser invocada através de um método padrão, abstraindo as especificidades do sensor;
   - Tratamento do Sensor - é responsável por encapsular a frequência de leitura de um sensor. Por exemplo, um determinado sensor deve ser lido a cada 20 segundos, já outro sensor a cada 90 segundos;
   - Tratamento de Dado Adquirido - atua como um filtro repassando somente as informações relevantes. Age selecionando informações que obedecem a uma determinada flutuação de valor e/ou estejam compreendidas em uma determinada faixa;
   - Módulo de Persistência - responsável por armazenar informações adquiridas em um banco de dados local, funciona como um mecanismo de tolerância a falhas caso haja problemas de transmissão dos dados para os outros módulos do uMED;
   - Módulo de Publicação - é responsável pela transmissão de dados adquiridos ao Servidor de Contexto;
   - Leitura do Status dos Sensores - verificação do status dos diferentes sensores considerando os seus aspectos operacionais.

2. Módulo Atuação

   O módulo Atuação do Gerente de Borda possui várias funções que se dividem conforme destacado a seguir:

   - Driver do Atuador - encapsula o tratamento de baixo nível dos atuadores de modo a facilitar a inserção de novos atuadores no Gerente de Borda, este driver implementa uma interface padrão. Desta forma abstrai as especificidades dos atuadores;
   - Tratamento do Atuador - é responsável por encapsular políticas de trabalho dos atuadores. Os atuadores realizam ações sobre o meio físico. Estas ações são regidas por políticas e parâmetros como: tipo de ação (ex. ligar, desligar), tempo de duração (ex. ligar por 60 segundos), potência (ex. ligar na velocidade máxima), entre outras;
   - Leitura do Status dos Atuadores - captura do estado operacional dos diferentes atuadores.

Repositório do Gerente de Borda

O Gerente de Borda contempla um repositório denominado RTDC (Repositório Temporário de Dados Coletados) objetivo deste repositório é atuar registrando (bufferizando), todos os dados coletados pelos sensores para o envio destes ao Servidor de Contexto.

Fluxos de Informações do Gerente de Borda

Neste seção é apresentado os fluxos pertinentes ao Gerente de Borda (vide figura 6.1).

O Servidor de Contexto repassa parâmetros operacionais ao Gerente de Borda através de fluxo que é denominado "Requisição/Configuração" trafegam parâmetros operacionais necessários para configuração de sensores que serão utilizados em um determinado contexto ou na requisição de rotina de verificação do status do sensor e/ou atuador.

Por sua vez, o fluxo do Gerente de Borda para o Servidor de Contexto cuja denominação é "Publicação" trafegam informações obtidas com a verificação (leitura) do status do sensor.

O fluxo de informação entre o Gerente de Atuação e o Gerente de Borda que é denominado "Comando de Atuação" foi discutido na seção 6.1.1.

6.1.4 Gerente de Comunicação

O Gerente de Comunicação é responsável pelo envio de mensagens *off-line* e *on-line*. Com este serviço será possível notificar os profissionais de saúde ou opcionalmente familiares do paciente caso ocorra alguma emergência em uns dos pacientes que estão sob monitoramento.

Está previsto que estejam disponíveis aplicações para realização de configurações no Gerente de Comunicação, como por exemplo, definir o envio dos tipos de mensagens, ou até mesmo a supressão de um determinado tipo. As mensagens podem ser de três tipos:

1. Mensagens instantâneas através da rede celular, SMS (Short Message Service);
2. Mensagens através do Gtalk (Google Talk);
3. Mensagens tipo emails padrão da *Internet*.

Está previsto em caso de falha no envio de mensagens ao profissional de saúde responsável, que o administrador da aplicação seja notificado através de um SMS. Estas mensagens que por algum motivo falharam devem ser registradas com "Status Pendente" no Repositório de Comunicações. O mesmo está sendo previsto para as notificações que obteram êxito no envio, estas serão registradas no RC (Repositório de Comunicações) com "Status Sucesso". Isto permitirá ao administrador da aplicação emitir o relatório STATUS DE MENSAGEM ENVIADAS (vide seção 6.1.2). Salvo as mensagens enviadas através da rede celular, algumas operadoras de telefonia celular não oferecem a possibilidade de confirmação de entrega de mensagens ao destinatário em um tempo hábil, ainda mais, se tratando de mensagens enviadas para operadoras diferentes da operadora de origem. Já no que diz respeito as mensagens enviadas para Google Talk ou email é

possível registrar o status de envio, inclusive no caso do Google Talk seria possível identificar se o usuário está on-line ou não, já para o email o registro seria de sucesso, salvo quando houvesse algum falha no provedor de email ou determinado endereço de email não existir.

Mensagens off-line

O Gerente Comunicação envia mensagens do tipo *off-line* entende-se por esta mensagem aquela em que o destinatário não precisa estar conectado no momento do envio, o exemplo deste tipo de mensagem e o email. Na figura 6.6 pode se observar um fluxo de sequência de envio de mensagens *off-line* disparada manualmente.

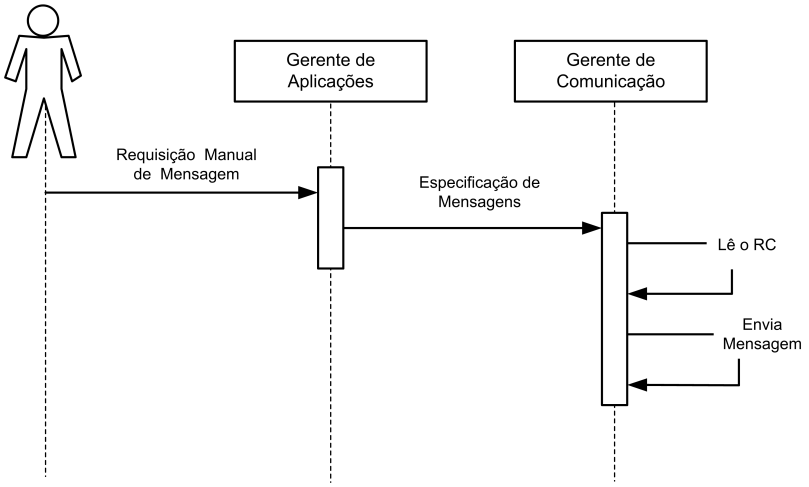

Figura 6.6: Diagrama de Sequência de Mensagem Off-line Disparada Manualmente

Mensagens on-line

O Gerente Comunicação envia mensagens do tipo *on-line* entende-se por esta mensagem aquela em que o destinatário precisa estar conectado no momento do envio, o exemplo deste tipo de mensagem e o Gtalk e o SMS. Na figura 6.7 pode se observar um fluxo de sequência de envio de mensagens *on-line* disparada pelo contexto.

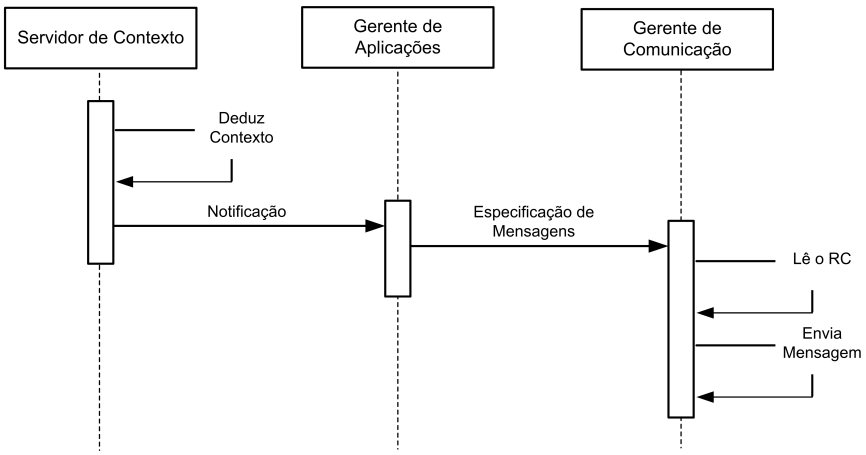

Figura 6.7: Diagrama de Sequência de Mensagem On-line Disparada pelo Contexto

Repositório do Gerente de Comunicação

O Gerente de Comunicação contempla um repositório denominado de RC, sendo o responsável por armazenar informações de cadastro dos profissionais de saúde envolvidos, assim como o cadastro dos familiares responsáveis pelo paciente. Estas informações são indispensáveis para gerência do envio de alertas. As informações tratadas são: nome, número do telefone celular, email, conta do Gtalk, entre outras. Neste repositório também ficam armazenadas todas as mensagens disparadas por este Gerente. As mensagens podem ter dois estados: (i) "Status Sucesso" para mensagens que obteram êxito no envio ou, (ii) "Status Pendente" para as mensagens que não tiveram confirmação de entrega para o destinatário.

Fluxos de Informações do Gerente de Comunicação

Neste seção é apresentado o fluxo pertinente ao Gerente de Comunicação (vide figura 6.1).

O fluxo entre o Gerente de Aplicações e o Gerente de Comunicação é denominado "Especificação de Mensagens" (fluxo é bidirecional) este fluxo foi discutido na seção 6.1.2.

6.1.5 Servidor de Contexto

Este servidor atua coletando informações e promovendo as correspondentes ações com relação ao ambiente ubíquo. Uma descrição dos gerentes que compõem o Servidor de Contexto é feita a seguir. O uMED promoveu uma versão atualizada do EXEHDA-SS proposto em (VENECIAN, 2010).

- Gerente de Aquisição

  O Gerente de Aquisição tem como função principal prover a captura de informações de contexto, disponibilizando as mesmas em um formato adequado para que o Gerente de Interpretação possa implementar suporte semântico utilizando os mesmos. Para a aquisição de contextos através de sensores e publicação de dados a partir dos mesmos se faz necessário: (i) especificar intervalos de tempo entre medições; (ii) flutuação mínima para que aconteça a publicação; (iii) definir a faixa na qual os valores dos sensores deverão ser publicados.

  Neste sentido, foi definido na OntContext, a classe *Contexto_Interesse*, onde cada contexto de interesse possui por aplicação, uma relação de sensores com seus parâmetros operacionais, regras para dedução, tradução das informações coletadas por estes sensores e o máximo de instâncias armazenadas dos sensores no RIC (Repositório de Informações Contextuais).

  Quando de uma publicação, o tradutor e instanciador contextual presentes neste gerente, identifica os contextos de interesse pertencentes ao sensor em execução, processa as regras de tradução definidas pelo programador da aplicação e armazena no RIC, representado ontologicamente pela classe *Contexto*.

  No RIC são instanciadas as informações coletadas dos sensores, tais como: identificador do sensor, valor coletado pelo sensor, valor que tenha ocorrido tradução

contextual, contexto de interesse pertencente ao sensor, nome do nodo, usuário e horário da coleta.

Portanto, após obter os contextos de interesse pertencentes a uma determinada publicação, o Gerente de Interpretação é acionado disponibilizando os contextos de interesse para a execução no Motor de Inferência.

- Gerente de Interpretação

  Três são as principais atribuições para este gerente: (i) manter o RIC, que armazena os contextos capturados pelo Gerente de Aquisição; (ii) utilizar um Motor de Inferência para processamento e dedução sobre as informações de contexto mantidas no RIC e nos Contextos de Interesses das Aplicações; (iii) alimentar o RCN (Repositório de Contexto Notificado), que armazena os estados dos contextos disponibilizados pelo Gerente de Notificação.

  O Motor de Inferência, identifica os sensores e as regras definidas pertencentes ao contexto de interesse repassado pelo Gerente de Aquisição e executa dois possíveis processamentos:

  – leitura no RIC dos valores coletados pelos sensores envolvidos do respectivo contexto de interesse em processamento - inferência baseada em Ontologias - Máquina de Inferência *OWLReasone* (JENA, 2010). Caso ainda não tenha ocorrido uma publicação de informação contextual pelos sensores envolvidos do respectivo contexto de interesse, é passado o valor *default* do sensor, definido na classe Sensor_Public;

  – dedução de regras definidas pelo desenvolvedor na classe *Contexto_Deduzido* da OntContext produzindo desta forma contextos deduzidos - inferência baseada em regras - Máquina de Inferência *Generic rule reasoner* (JENA, 2010).

  Por fim, o processo, o Gerente de Interpretação, instância no RCN do Gerente de Notificação, os valores dos contextos e as deduções processadas pelo Motor de Inferência. A figura 6.8 apresenta o fluxo de interpretação contextual do motor de inferência do EXEHDA-SS.

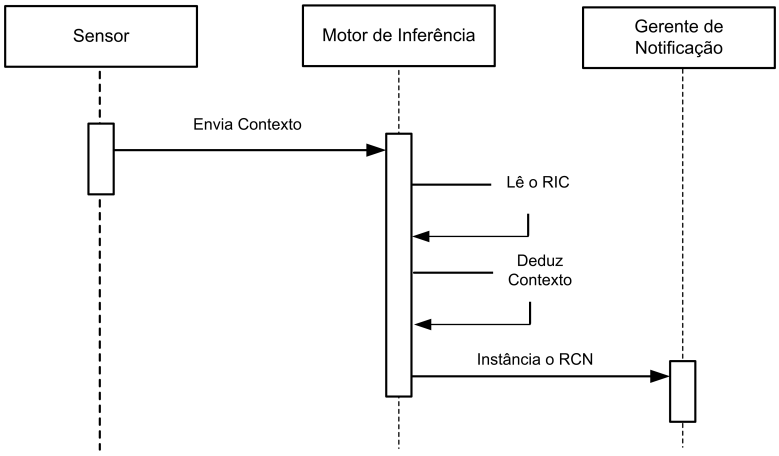

Figura 6.8: Fluxo do Motor de Inferência do EXEHDA-SS (VENECIAN, 2010)

- Gerente de Notificação

  Esse gerente é responsável por disponibilizar os contextos processados pelo Gerente de Interpretação. Duas são as principais atribuições deste gerente: (i) notificação de informações contextuais ao serviço de adaptação dinâmica do EXEHDA - EXEHDA-DA (WARKEN, 2010); (ii) receber subscrições pelas aplicações e notificá-las de acordo com as solicitações realizadas.

  A notificação para o serviço de adaptação dinâmica do *middleware* EXEHDA ocorre através do envio da instância do contexto notificado ao mesmo. O serviço de adaptação ao receber a instância que contém alterações contextuais no ambiente ubíquo, realiza uma leitura dos demais valores no contexto notificado da OntContext e inicia suas tarefas de execução previstas em seu processamento.

  Uma outra funcionalidade disponível no EXEHDA-SS é que as aplicações podem realizar subscrições no Gerente de Notificação. A aplicação deve invocar o método *ExehdaSS_Subscricao(app,userId,idSubscricao)*, passando três parâmetros: código da aplicação, usuário e o identificador do nodo que solicitou a subscrição. O subscritor presente no Gerente de Notificação, recebe esses parâmetros e dispara uma leitura no RCN encaminhando para a aplicação as informações da respectiva subscrição (vide figura 6.9). Esta forma de execução definida para o Subscritor faz com que o serviço de sensibilidade ao contexto do EXEHDA-SS possa ser utilizado diretamente pelas aplicações.

```
<?xml version="1.0" enconding="UTF-8"?>
<EXEHDA-SS>
<Contexto_Notificado UsuarioId="AFD10-4589E">
<Aplicacao AplicacaoId="100">
<Sensor SensorId="100" SensorDesc="Sensor de Frequência Cardíaca" SensorValor="180" />
<Sensor SensorId="101" SensorDesc="Sensor de Temperatura Corporal" SensorValor="28°" />
<Sensor SensorId="102" SensorDesc="Sensor de Pressão Arterial" SensorValor="190110" />
</Aplicacao>
</Contexto_Notificado>
</EXEHDA-SS>
```

Figura 6.9: Exemplo de Notificação às Aplicações (VENECIAN, 2010)

Um aspecto importante a ser destacado é que o Servidor de Contexto pode ter as regras de tratamento contextual modificadas durante a execução. Dentre outros aspectos este perfil operacional garante uma operação ininterrupta, o que se mostra oportuno, considerando que um mesmo Servidor de Contexto pode atender simultaneamente a várias aplicações.

Repositório do Servidor de Contexto

O Servidor de Contexto contempla dois repositório de dados um denominado de RIC e o outro RCN os mesmos já foram discutidos nesta mesma seção.

Fluxos de Informações do Servidor de Contexto

Os fluxos que envolvem o Servidor de Contexto foram discutidos nas seções que abordaram os gerentes do uMED, neste mesmo capítulo.

6.2 Considerações sobre o Capítulo

Neste capítulo foram apresentado os critérios de concepção do uMED. Foi discutida a modelagem de sua arquitetura de *software*, sendo detalhados seus módulos: Gerente de Atuação, Gerente de Aplicações, Gerente de Borda e o Gerente de Comunicação. Também foi contemplada nesse capítulo uma descrição do Servidor de Contexto incorporado ao uMED. O próximo capítulo, apresenta as principais tecnologias utilizadas no uMED e um estudo de caso.

7 UMED: TECNOLOGIAS UTILIZADAS E ESTUDO DE CASO

Neste capítulo estão resumidos os principais aspectos das tecnologias utilizadas, bem como o estudo de caso empregado na avaliação das funcionalidades do uMED. O estudo de caso contemplou tarefas referentes a coleta de informações contextuais, seu tratamento e as respectivas atuações no ambiente. Este capítulo contempla duas subseções, a primeira apresentando as principais tecnologias utilizadas e a segunda o estudo de caso.

7.1 Principais Tecnologias Utilizadas

Esta subseção apresenta as principais tecnologias empregadas na concepção do uMED. Dentre estas destaca-se: (i) linguagem Java; (ii) JSF um framework voltado ao desenvolvimento de aplicações em JAVA; (iii) PostgreSQL; (iv) Web Service; (v) XML ; (vi) DB4O; (vii) YAML. A seguir uma breve caracterização dos aspectos entendidos como importantes nas mesmas, considerando as características propostas para o uMED.

7.1.1 Java

A independência de plataforma que Java oferece, a grande quantidade de bibliotecas disponíveis e a existência de máquinas virtuais embutidas em vários dispositivos tornou a linguagem Java uma tecnologia chave para o desenvolvimento de *software* na computação ubíqua.

Java originou-se como parte de um projeto de pesquisa que visava a criação de um *software* avançado que atendesse a uma extensa variedade de maquinário de redes e sistemas embutidos. O objetivo inicial era desenvolver um ambiente operacional pequeno, confiável, portável e distribuído. Inicialmente, a linguagem escolhida foi C++. Porém, com o passar do tempo, as dificuldades encontradas com C++ aumentaram até o ponto em que os problemas poderiam ser melhor endereçados se fosse criada uma linguagem completamente nova (JAVA, 2010).

Deste modo, Java foi projetada para atender a vários requisitos desejáveis em uma linguagem de programação, como por exemplo, confiabilidade, devido ao seu gerenciamento de memória, o que resulta em um ganho de redigibilidade e reuso de código (JAVA, 2010).

7.1.2 JSF

O JSF (*JavaServer Faces*) é um *framework* para o desenvolvimento de aplicações *Web*, que permite o desenvolvimento de aplicações para a *Internet* de forma visual. O JSF é atualmente considerado pela comunidade Java como um ótimo recurso em termos de desenvolvimento de aplicações *Web* utilizando Java (JSF, 2010).

Dentre as suas características importantes destacam-se: (i) fornece um conjunto de *tags* JSP (*Java Server Pages*) para acessar os componentes; (ii) reutiliza componentes da página; (iii) associa os eventos do lado cliente com os manipuladores dos eventos do lado do servidor, neste sentido os componentes de entrada possuem um valor local representando o estado no lado servidor.

7.1.3 PostgreSQL

O PostgreSQL (POSTGRESQL, 2010) é um SGBDOR (Sistema Gerenciador de Banco de Dados Objeto-relacional) de código aberto coordenado pelo PostgreSQL *Global Development Group*. É fruto de uma ampla evolução iniciada na Universidade de Berkeley, Califórnia, com um projeto denominado Ingres. O seu código está sob a licença BSD (*Berkeley Software Distribution*), após uma série de mudanças significativas, o projeto foi renomeado para PostgreSQL. Desde então, um grupo de desenvolvedores e de voluntários de todo o mundo, coordenados através da *Internet*, mantêm o *software* e desenvolve novas funcionalidades.

Sendo apontado com um dos SGBDs (Sistema Gerenciador de Banco de Dados) de código aberto mais avançados da atualidade, o PostgreSQL possui uma arquitetura que ao longo do tempo ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. Além de ser compatível com os principais sistemas operacionais, incluindo GNU/Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e MS Windows (POSTGRESQL, 2010).

A aceitação do PostgreSQL tem se ampliado para além da comunidade de código aberto. Há entre os seus usuários, grandes empresas internacionais, órgãos governamentais de vários países e universidades de prestígio mundial.

7.1.4 Web Service

O Web Service é uma arquitetura de *software* projetada para suportar interação máquina-máquina através de uma infraestutura de comunicação. Ele tem uma interface descrita em um formato processável por máquina (especificamente WSDL - *Web Services Description Language*). Outros sistemas interagem com o Web Service de uma maneira prescrita por sua descrição usando mensagens SOAP (Simple Object Access Protocol), normalmente transmitida através de HTTP (*Hypertext Transfer Protocol*) com uma serialização XML em conjunto com outros padrões relacionados à Web (W3C, 2010).

Essencialmente, o Web Service faz com que os recursos de *software*, independente da tecnologia de implementação utilizada, estejam disponíveis de uma forma normalizada (WSDL). Outras tecnologias fazem o mesmo, como por exemplo, os *browsers* da *Internet*, o qual acedem às páginas Web disponíveis usando por norma as tecnologias da *Internet*, HTTP e HTML (*HyperText Markup Language*). No entanto, estas tecnologias não são bem sucedidas na comunicação e integração de aplicações. Sendo assim, existe uma grande motivação sobre a tecnologia Web Service pois possibilita que diferentes aplicações comuniquem entre si e utilizem recursos distintos.

Os Web Services são identificados por um URI (*Uniform Resource Identifier*), descritos e definidos usando XML. Um das razões que tornam os Web Services atrativos é o fato deste modelo ser baseado em tecnologias *standards*, em particular XML e HTTP. Nesta perspectiva, os Web Services são utilizados, principalmente, para disponibilizar serviços interativos na Web. (W3C, 2010).

7.1.5 XML

XML é uma linguagem de marcação recomendada pela W3C. Em meados da década de 1990, a W3C começou a trabalhar em uma linguagem de marcação que combinasse a flexibilidade da SGML (*Standard Generalized Markup Language*) com a simplicidade da HTML. O XML é um formato para criação de documentos com dados organizados de forma hierárquica (XML, 2010).

Uma característica do XML é sua extensibilidade: é possível escrever suas próprias *tags* para descrever o conteúdo de um específico tipo de texto, por exemplo, bem como definir *schemas* que descrevem a estrutura de um tipo particular de documento XML.

No XML, são especificados em *schemas* quais *tags* podem ser usadas e onde elas podem ocorrer. Assim, diz-se que todo documento XML que segue essas especificações está conforme o determinado *schema*. Além disso, XML não inclui instruções de formatação/visualização. Assim, mantendo os dados separados das instruções de apresentação, podem-se expor os mesmos dados de diferentes maneiras (XML, 2010).

7.1.6 DB4O

Em um banco de dados relacional, os dados são armazenados no formato de linhas e colunas. No entanto, linguagens como Java, tratam dados numa abordagem de objetos. Desta forma, quando se trabalha com estas duas tecnologias, se faz necessário um processo de tradução de linhas e colunas para objetos e vice-versa (vide figura 7.1). Isto implica tanto em uma perda de produtividade de desenvolvimento, bem como de desempenho da aplicação pela execução das traduções que se fazem necessárias. Neste sentido, pode-se utilizar a tecnologia de banco de dados orientados a objetos, a qual permite persistir informação diretamente na forma de objetos, dispensando o processo de tradução (DB4O, 2010).

Segundo Zyl (ZYL; KOURIE; BOAKE, 2006), o *overhead* de tradução objeto-relacional implica que as implementações baseadas em banco de dados objeto-relacional sejam mais lentas que as construídas sobre um banco de dados orientado a objetos.

O DB4O é um banco de orientado a objetos, *open source* que possibilita aos desenvolvedores Java e .Net reduzir o tempo e o custo de desenvolvimento e alcançar altos níveis de performance. Foi desenvolvido para prover um completo banco de dados embutido para equipamentos, dispositivos móveis, plataformas *desktop* e servidores em ambientes orientados a objetos (DB4O, 2010).

7.1.7 YAML

YAML é um acrônimo recursivo que significa "*YAML Ain't Markup Language*" no início do seu desenvolvimento YAML significava "*Yet Another Markup Language*" para distinguir seu propósito centrado em dados no lugar de documentos marcados. Como é usado frequentemente XML para serialização de dados e XML é uma autêntica linguagem

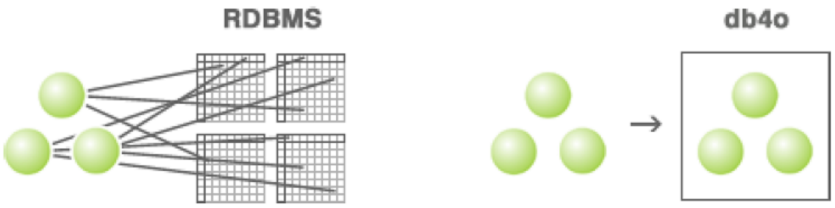

Figura 7.1: Bancos Objeto-Relacional e Banco de Objetos db4o (DB4O, 2010)

de marcação de documentos, é razoável considerar o YAML como uma linguagem de marcação rápida.

O YAML (BEN-KIKI; EVANS, 2005) é um formato de serialização de dados legível por humanos, que se inspirou em conceitos de linguagens como o XML, C, Python, Perl e também o formato de correio eletrônico especificado no RFC (Request for Comments) 2822.

Segundo Fonseca (FONSECA; SIMÕES, 2007), o YAML é em grande parte mais simples de ler, editar, modificar e produzir que o XML. O autor destaca que quase tudo o que é possível de representar em XML pode ser representado em YAML, e ao mesmo tempo, de uma forma mais compacta. Existem bibliotecas YAML para quase todas as linguagens de programação.

7.2 Estudo de Caso - Acompanhamento Clínico de Pacientes

Este estudo de caso consiste em uma aplicação sintética direcionada à área de Medicina Ubíqua denominada ACP (Acompanhamento Clínico de Pacientes), cujas funcionalidades destacadas nesta seção foram concebidos com o objetivo de explorar o suporte oferecido pelo uMED.

Os valores correspondentes aos dados sensorados foram obtidos através de um *software* simulador que efetuava publicações no Servidor de Contexto do uMED. Tanto os valores praticados, quanto os níveis de alerta, bem como as regras são aproximações médias sem compromisso de traduzir uma realidade médica precisa.

7.2.1 Objetivos da ACP

A premissa buscada é qualificar o acompanhamento clínico do pacientes, que não estejam internados em UTI (Unidade de Tratamento Intensivo). Particularmente, com o intuito de aumentar o período de tempo que o paciente permaneça sob cuidado, e evitar deslocamentos desnecessários pelos profissionais de saúde para saber o estado em que se encontra um paciente. Nesta perspectiva, os objetivos contemplados na aplicação ACP são:

- Exibir dados de pacientes adquiridos dinamicamente por mecanismo de coleta de sinais vitais;
- Emitir, de forma automatizada, diferentes níveis de alertas, em função dos dados coletados, para os profissionais de saúde;

- Integrar o serviço de alertas da aplicação a rede aberta de comunicação *Google Talk*, SMS e email;
- Prover possibilidade de uso, tanto a partir de dispositivo móveis, como de mesa;
- Permitir acesso ubíquo ao histórico dos dados coletados dos pacientes pelos profissionais de saúde;
- Permitir acesso ubíquo ao histórico dos dados de atuação nos pacientes pelos profissionais de saúde;
- atuar remotamente em dispositivos médicos que estejam associados ao paciente.

A aplicação ACP está organizada para ser integrada ao Sistema de Informações do Hospital. Nos testes foi utilizada uma base de dados sensoriados criada especificamente para esta etapa do trabalho. Sobre esta base criada pelo G3PD foi possível ter liberdade no que diz respeito à manipulação dos dados utilizados.

As informações cadastrais são de quatro naturezas: (i) dados cadastrais de pacientes; (ii) dados dos profissionais de saúde; (iii) dados aquisitados dinamicamente pela rede de sensores de sinais vitais de pacientes; (iv) dados de ativação de atuadores.

Na versão da ACP utilizada para avaliação do uMED, os profissionais de saúde podem ser tanto enfermeiros quanto médicos.

Quando necessário, os mesmos podem disparar o Módulo Aquisição do Gerente de Borda e capturar dinamicamente os dados correspondentes aos sinais vitais dos pacientes, bem como, ativar o Módulo Atuação do Gerente de Borda e atuar remotamente em dispositivos associados ao tratamento do paciente.

Os sinais capturados para este estudo de caso são frequência cardíaca, pressão arterial e temperatura corporal. Em função dos valores dos sinais coletados, são produzidos diferentes níveis de alertas aos profissionais de saúde, por sua vez, dependendo do nível de alerta é disponibilizada a opção de envio de mensagens aos profissionais de saúde ou, ainda, no caso de alerta máximo, o envio incondicional de mensagens.

Por sua vez, os dispositivos médicos que atuaram junto aos pacientes são aparelho de oxigênio, bomba de infusão, ventilador pulmonar (simulados). Em função das políticas de atuação os dispositivos são configurados para exercer suas funções obdecendo a parâmetros como potência de atuação, tempo de atuação, entre outras.

7.2.2 Monitoramento Pró-ativo de Pacientes

Na aplicação ACP (vide figura 7.2) é possível observar últimos alertas disparados, a representação dos níveis de alerta são classificados pela cor do alerta, conforme destacado a seguir:

- Alerta Verde: sinais vitais normais - indica que todos os sinais monitorados entre eles: frequência cardíaca, pressão arterial e temperatura corporal se encontram na faixa aceitável;
- Alerta Amarelo: sinal vital irregular - indica que um dos sinais monitorados se encontra em uma faixa irregular;
- Alerta Vermelho: sinais vitais críticos - indica que dois ou mais sinais monitorados se encontram em uma faixa irregular, este alerta é considerado o alerta máximo.

Figura 7.2: Monitoramento Pró-ativo de Pacientes

7.2.3 Coleta Pró-ativa de Sinais Vitais

Na aplicação ACP (vide figura 7.3) é possível configurar os sensores associados ao paciente nesta interface pode-se selecionar o tipo de sensor, bem como, determinar os seus parâmetros operacionais.

7.2.4 Geração de Relatórios

Uma das funcionalidades previstas pelo uMED é possibilidade de criação de relatórios personalizáveis pelos usuários do sistema, entre eles: (i) MONITORAMENTO; (ii) ATUAÇÃO; (iii) HISTÓRICO CLÍNICO. A seguir estão resumidas as funcionalidades e suas respectivas interfaces para cada um dos relatórios citados:

Relatórios de Monitoramento

Neste relatório de MONITORAMENTO será facultado ao usuário da aplicação obter uma listagem dos dados coletados. Este tipo de relatório podem ser personalizado através de duas formas gerais: (i) por período de tempo; (ii) através de faixa de valores do dispositivo. Na figura 7.4 é apresentada a interface correspondente ao relatório de MONITORAMENTO por faixa de valores do dispositivo.

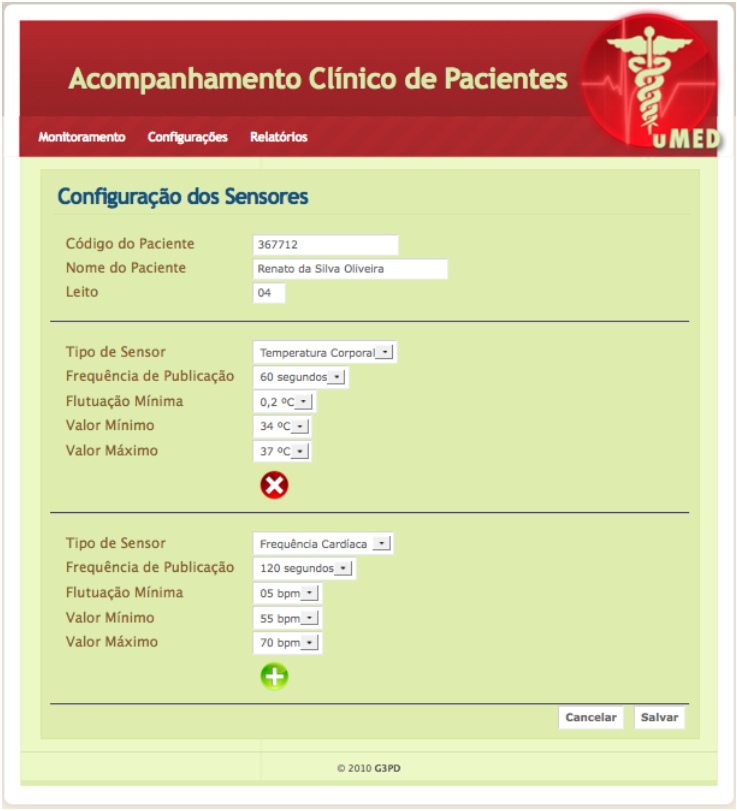

Figura 7.3: Configuração dos Sensores

Figura 7.4: Relatório de Monitoramento por Faixa de Valores do Dispositivo

Relatórios de Atuação

No relatório de ATUAÇÃO será concedido ao usuário da aplicação uma listagem referente aos dados que foram utilizados pelos atuadores. Estes tipos de relatórios podem ser personalizados através de duas formas gerais: (i) por período de tempo; (ii) através de faixa de valores do dispositivo. Na figura 7.5 é apresentada a interface correspondente ao relatório de ATUAÇÃO por período de tempo.

Figura 7.5: Relatório de Atuação por Período de Tempo

Geração de Dados para Estudo de Caso

Com as informações armazenadas dos sensores e atuadores é possível a criação de relatório de HISTÓRICO CLÍNICO proporcionando um estudo mais aprofundado do caso clínico do paciente.

Estes relatórios de históricos poderiam ser utilizados como material didático destinado a formação dos profissionais de saúde, ou até mesmos, para discussão de casos clínicos pelas juntas médicas, possibilitando assim segunda opinião de outros profissionais de saúde. Na figura 7.6 é apresentada a interface correspondente ao relatório de HISTÓRICO CLÍNICO por período de tempo.

7.2.5 Atuação Pró-ativa de Pacientes

Na aplicação ACP (vide figura 7.7) é possível configurar um dispositivo atuador associado ao paciente. Nesta interface pode-se selecionar o tipo de dispositivo, bem como, determinar sua política operacional de atuação.

7.2.6 Configuração de Envio de Notificações

Na aplicação ACP (vide figura 7.8) para cada nível de alerta é possível configurar o envio de notificações aos profissionais de saúde. Existem três tipos de notificações, sendo que no alerta vermelho é obrigatória a escolha de pelo menos um tipo.

Figura 7.6: Relatório de Histórico Clínico

Figura 7.7: Configuração dos Atuadores

Figura 7.8: Configuração de Envio de Notificações

7.3 Considerações sobre o Capítulo

Este capítulo apresentou as principais tecnologias empregadas na concepção do uMED e um estudo de caso com aplicações voltadas para area médica, entre eles: (i) o monitoramento de pacientes com a possibilidade de emissão de diversos tipos de alerta; (ii) e atuação em dispositivos hospitalares; (iii) a possibilidade de criação de relatórios personalizáveis pelos usuário do sistema; (iv) o aproveitamento dos dados armazenados para estudo de casos clínicos de pacientes. O próximo capítulo, apresenta as considerações finais, as principais contribuições, as publicações realizadas e trabalhos futuros, associados ao uMED.

8 CONSIDERAÇÕES FINAIS

Neste capítulo são resumidos os principais aspectos decorrentes do desenvolvimento deste trabalho. São discutidas as principais contribuições do trabalho, são relacionadas as publicações realizadas, e, por fim, são apresentados possíveis trabalhos futuros.

A computação ubíqua vem mostrando potencial para atender necessidades de diversas áreas do mundo real. Um setor importante que as tecnologias ubíquas têm sido aplicadas é o domínio dos cuidados com a saúde. Como citado previamente, em ambientes de saúde, os profissionais de saúde (médicos e enfermeiros) estão constantemente em movimento, seja deslocando-se entre setores de hospitais ou em ambientes externos aos mesmos. O escopo dessa proposta é voltado para auxiliar estes profissionais de saúde em sua rotina diária nos hospitais, no que diz respeito aos cuidados de saúde de pacientes que necessitam de supervisão constante. A rotina destes profissionais envolve nomadismo e fragmentação de atividades, conforme apontado nos capítulos iniciais desta dissertação.

Uma das formas de prestar suporte ao nomadismo dos profissionais de saúde é permitir que ele continue executando suas atividades, sabendo que determinados pacientes estão sobre monitoramento e as informações decorrentes estão disponíveis de qualquer lugar bastando para isso ter acesso a um dispositivo computacional, como por exemplo, PDAs, *Notebooks, e Desktops* entre outros. Sendo assim, este profissional de saúde não necessariamente precisa se deslocar até determinado paciente para saber o estado de saúde que ele se encontra.

Com o objetivo de colaborar para superar alguns desafios da área de medicina ubíqua, essa dissertação está propondo uma arquitetura sensível ao contexto que gerencia sinais vitais de pacientes, com possibilidade de alteração de regras de monitoramento em tempo de execução. Entende-se por tempo de execução a flexibilidade da arquitetura de *software* em reconfigurar dinamicamente as regras para manipulação e processamento das informações contextuais.

Entende-se que o uso de tal arquitetura em um sistema de apoio a vida irá possibilitar uma diminuição no nomadismo e fragmentação de atividades exercidas pelos profissionais de saúde e com isso a possibilidade de um ganho de produtividade, já que haverá menos investimento de tempo em deslocamentos.

Neste sentido o uMED contempla além do monitoramento dos sinais vitais do paciente e/ou condições do seu ambiente (temperatura, umidade), também a possibilidade de gerar alertas conforme regras definidas pelos próprios profissionais de saúde, esses alertas, por exemplo, podem ser a ativação de uma campainha em uma enfermaria do hospital e/ou até mesmo o envio de um SMS ao médico responsável.

Outra funcionalidade da arquitetura do uMED é a possibilidade de intervenção,

ou seja, de atuação no ambiente. A partir dos dados obtidos através do monitoramento o profissional de saúde pode decidir a partir dos mesmos, e promover o acionamento, configuração ou desligamento de algum equipamento, como por exemplo, umidificadores, aquecedores, alarmes, luzes de sinalização e até mesmo dipositivos com relação com demandas médicas: aparelho de oxigênio, bomba de infusão, ventilador pulmonar, entre outros.

A arquitetura do uMED possibilita a criação de relatórios personalizáveis pelos próprios profissionais de saúde. Estes relatórios podem ser utilizados para uma futura análise contribuindo para o estudo do caso clínico do paciente.

8.1 Principais Contribuições

As atividades desenvolvidas ao longo deste trabalho, tendo por base os estudos realizados, na construção de uma arquitetura direcionado à medicina ubíqua, conduziram a obtenção das seguintes contribuições: A seguir é apresentada uma tabela 8.1 com as principais contribuições decorrentes deste trabalho e pautando-as com os trabalhos relacionados já discutidos no capítulo 4.

Considerando o elencado na tabela 8.1 de forma sucinta podemos dizer que a contribuição central do trabalho realizado contemplou um *framework* direcionado a aplicações de medicina ubíqua, que trata de modo sinérgico quatro frentes: (i) uma relacionada a aquisição de dados monitorados; (ii) outra pertinente a atuação no ambiente ubíquo; (iii) uma que diz respeito a gerência dos diferentes níveis de alerta conforme regras contextuais processadas pelo *framework* uMED; (iv) a geração de dados para estudo dos casos clínicos.

8.2 Publicações Realizadas

- CIC-UCPEL 2009: RODRIGUES, Sérgio, YAMIN, Adenauer C. Sistematizando Desafios de Pesquisa em Medicina Ubíqua. In. 18ª Congresso de Iniciação Científica da Universidade Católica de Pelotas. Pelotas, RS. 8ª Mostra de Pós-Graduação, 2009.

- ERAD 2010: RODRIGUES, Sérgio, YAMIN, Adenauer, AUGUSTIN, Iara. Revisando o Desenvolvimento de Aplicações na Medicina Ubíqua. In 10ª Escola Regional de Alto Desempenho, 2010, Passo Fundo-RS. ERAD 2010 - Fórum de Pós-Graduação da 10ª Escola Regional de Alto Desempenho, 2010. p.137-138.

- CBIS 2010: RODRIGUES, Sérgio, VENECIAN Luthiano. uMED - Uma Proposta Baseada em Processamento Semântico para Medicina Ubíqua. In XII Congresso Brasileiro de Informática na Saúde CBIS, Porto de Galinhas-PE. CBIS 2010.

- CLCAR 2010: RODRIGUES, Sérgio, YAMIN, Adenauer, GEYER, Cláudio, LOPES, João, VENECIAN, Luthiano. Explorando o Processamento Semântico para Sensibilidade ao Contexto na Computação em Grade. Conferência Latino Americana de Computação de Alto Rendimento CLCAR, Gramado-RS. CLCAR 2010.

- ERRC 2010: RODRIGUES, Sérgio, VENECIAN, Luthiano, LOPES, João, YAMIN, Adenauer. uMED: Uma Arquitetura de Software Direcionada à Medicina

Tabela 8.1: Principais Contribuições do uMED

Contribuição do uMED	**Comparação com Trabalhos Relacionados**
Concepção de uma arquitetura que contempla cinco serviços cooperantes para atender as tarefas de monitoramento de dados contextuais, de atuação no meio e emissão de alertas aos profissionais de saúde;	Nenhum dos trabalhos relacionados contempla uma arquitetura com cinco serviços a maioria são de serviços individualizados;
Proposição para a arquitetura concebida de suporte para sensibilidade ao contexto, tendo sido modelada para ser expansível, tanto no que diz respeito a captura de dados do ambiente ubíquo, bem como, quanto aos possíveis consumidores de contextos de interesse. Os consumidores podem ser as aplicações que se subscrevem no servidor de contexto;	Nos projetos ABC, Awareness, UbiDoctor e ClinicSpace é contemplado esta característica;
Elaboração de mecanismo para a criação de relatórios personalizáveis pelos usuários do sistema, possibilitando a geração de dados para estudo dos casos clínicos;	Nenhum dos trabalhos relacionados contempla esta característica;
Criação de estratégias que permitam a inserção ou remoção de sensores e/ou atuadores de forma que sejam abstraídas as especificidades dos dispositivos;	Nenhum dos trabalhos relacionados contempla esta característica;
Disponibilização de um mecanismo de tolerância a falhas de comunicação nos Gerentes de Borda, o qual persiste os dados adquiridos pelos sensores, garantindo sua disponibilização ao Servidor de Contexto quando forem restabelecidas as comunicações;	No projeto Awareness existe um mecanismo de tolerância a falhas nos sensores;
Concepção de interface para os profissionais de saúde especificarem o recebimento de alertas através de mensagens, esta interface permite ao profissional de saúde determinar as suas preferências quanto aos tipos de mensagens;	No projeto ClinicSpace é possível o recebimento de alertas através de mensagens por email;
Especificação de características funcionais que permitam ao usuário realizar alterações de regras em tempo de execução;	No projeto ClinicSpace é possível alterar regras de monitoramento em tempo de execução;
Especificação de características funcionais que permitam ao usuário realizar leitura instantânea de sensores, utilizados nas aplicações;	Nos projetos ABC, Awareness e ClinicSpace esta característica esta presente;
Especificação de características funcionais que permitam ao usuário manipular de forma instantânea os atuadores, utilizados nas aplicações;	No projeto ClinicSpace esta característica esta presente;
Especificação de características funcionais que permitam ao usuário promover alteração nos parâmetros de configuração dos atuadores remotamente;	No projeto ClinicSpace esta característica esta presente;
Especificação de características funcionais que permitam ao usuário especificar os parâmetros operacionais para publicação dos sensores pertencentes as aplicações;	Nos projetos ABC, Awareness e ClinicSpace esta característica esta presente;
Especificação de características funcionais que permitam ao usuário especificar os parâmetros operacionais para configuração dos atuadores pertencentes as aplicações;	No projeto ClinicSpace esta característica esta presente;

Ubíqua. In 8ª Escola Regional de Redes de Computadores, Alegrete-RS. ERRC 2010 - Fórum de Pós-Graduação da 8ª Escola Regional de Redes de Computadores, 2010.

- CIC-UCPEL 2010: RODRIGUES, Sérgio, YAMIN, Adenauer C. Provendo Soluções Ubíquas para Medicina. In. 19ª Congresso de Iniciação Científica da Universidade Católica de Pelotas. Pelotas, RS. 9ª Mostra de Pós-Graduação, 2010. Prêmio Pesquisador em Pós-Graduação.

- MCSUL 2010: RODRIGUES, Sérgio, YAMIN, Adenauer, VENECIAN, Luthiano LOPES, João. uMED - Um Modelo para a Medicina Ubíqua Baseado em Processamento Semântico. In 4ª Southern Conference on Computational Modeling, 2010, Rio Grande-RS. MCSUL 2010 - Fórum de Pós-Graduação da 4ª Southern Conference on Computational Modeling, 2010. p.91-96.

- ERAD 2011: RODRIGUES, Sérgio, VENECIAN , Luthiano, DILLI, Renato, WARKEN, Nelsi, YAMIN, Adenauer. uMED: Sensibilidade ao Contexto na Medicina Ubíqua. In 11ª Escola Regional de Alto Desempenho, 2011, Porto Alegre-RS. ERAD 2011 - Fórum de Pós-Graduação da 11ª Escola Regional de Alto Desempenho, 2011. Aceito para publicação.

8.3 Trabalhos Futuros

Dentre os aspectos levantados para continuidade do trabalho destacamos:

- avaliar a possibilidade do retorno de uma "confirmação" para as mensagens notificadas através da rede de telefonia celular pelo Gerente de Comunicação. Esta qualificação do Gerente de Comunicação utilizado no uMED, está em processo de negociação ante as operadoras de telefonia celular. Na mesma direção consolidar este recurso na interface com o Gtalk;

- ampliar o estudo de padrões de sensores e atuadores no desenvolvimento de aplicações direcionados a área médica;

- empregar um sistema especialista no processamento de regras contextuais no Servidor de Contexto do *framework* uMED;

- reavaliar os protocolos e procedimentos a serem empregados na troca de informações entre os gerentes do uMED;

- definir um mecanismo de adaptação de interfaces para o Gerente de Aplicações potencializando o uso a partir de diferentes dispositivos;

- disponibilizar o uMED aos profissionais da área de saúde do Hospital Universitário São Francisco de Paula com intuito de monitorar sinais vitais de pacientes entre eles: temperatura corporal, pressão arterial e frequência cardíaca, este emprego do uMED, no período de teste, acontecerá paralelamente com as demais rotinas médicas de registro e acompanhamento já empregadas;

- analisar o desempenho do *framework* uMED e uma situação real, parametrizando resultados quantitativos da aplicação, como por exemplo: quantidade de médicos, pacientes, enfermeiros envolvidos, quantidade de informações de contextos capturadas, quantidade de inferências, entre outras.

REFERÊNCIAS

AUGUSTIN, I. **Abstrações para uma Linguagem de Programação Visando Aplicações Móveis em um Ambiente de Pervasive Computing**. 2004. 194p. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre, RS.

AUGUSTIN, I.; YAMIN, A.; SILVA, F. L.; FERREIRA, G. L.; RIZZETTI, T. A. Grade Computacional como Infraestrutura para a Computação Pervasiva Ubiqua. **Escola Regional de Alto Desempenho ERAD 2008**, [S.l.], Março 2008.

BARDRAM, J. Hospitals of the Future - Ubiquitous Computing Support for Medical Work in Hospitals. **2nd Workshop on Ubiquitous Computing for Pervasive Healthcare Applications**, Seattle, 2003.

BARDRAM, J. Activity-based computing for medical work in hospitals. **ACM Transactions on Computer-Human Interaction**, [S.l.], v.16, n.2, june 2009.

BARDRAM, J.; BOSSEN, C. Mobile Work - The Spatial Dimension of Collaboration at a Hospital. **Computer Supported Cooperative Work**, [S.l.], v.14, n.2, p.131–140, 2005.

BARDRAM, J. E. Applications of Context-Aware Computing in Hospital Work: Examples and Design Principles. **Proceedings of the 2004 ACM Symposium on Applied Computing**, [S.l.], p.1574–1579, 2004.

BARDRAM, J. E.; CHRISTENSEN, H. B. Pervasive computing support for hospitals: An overview of the activity-based computing project. **Personal Ubiquitous Comput**, [S.l.], v.6, n.1, p.44–51, 2007.

BEN-KIKI, B. I. O.; EVANS, C. **Yaml specification**. Disponível em: < http://yaml.org/spec/ >. Acesso em agosto de 2010.

BROWN, I.; ADAMS, A. The ethical challenges of ubiquitous healthcare. **International Review of Information Ethics**, [S.l.], v.8, n.54-59, Dezembro 2007.

CAVALCANTI, D. e. a. Issues in integrating cellular networks wlans, and manets: a futuristic heterogeneous wireless network. **IEEE Wireless Communications**, [S.l.], v.12, n.3, 2005.

CFM, . **Resolução 1.629/2002 do Conselho Federal de Medicina**. Disponível em: < http://www.arnaut.eti.br/ResoCFM.htm >. Acesso em março de 2010.

CFM, . **Resolução 1.821/2007 do Conselho Federal de Medicina**. Disponível em: < http://www.sbis.org.br/indexframe.html >. Acesso em maio de 2010.

CHALMERS, D. e. a. **Ubiquitous Computing**: Experience, Design and Science. Disponível em: < http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/index.html >. Acesso em novembro de 2009: [s.n.], 2006.

COSTA, C. A.; YAMIN, A. C.; GEYER, C. F. R. Toward a General Software Infrastructure for Ubiquitous Computing. **IEEE Pervasive Computing**, Los Alamitos, CA, USA, v.7, n.1, p.64–73, 2008.

COULORIS, G.; DOLLIMORE, J.; KINDBERG, T. (Ed.). **Distributed Systems - Concepts and Design**. [S.l.]: Addison Wesley, 2005. 657 - 719p. n.cap. 6.

DAVID, L. B. Ubiquitous Computing. **Emerging Technologies for Learning**, [S.l.], v.2, 2007.

DB4O. **db4obejcts**. Disponível em: < http://www.db4o.com/ >. Acesso em maio de 2010.

DINIZ, J. **UbiDoctor**: Arquitetura de Serviços para Gerenciamento de Sessão e Adptação de Conteúdo em Ambientes de Medicina Ubíqua. 2009. 178p. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Pernambuco, UFPE, Recife, PE.

FAVELA, J.; RODRIGUEZ, M.; PRECIADO, A.; GONZALES, V. M. Integrating Context-Aware Public Displays Into a Mobile Hospital Information System. **IEEE Transactions On Information Technology In Biomedicine**, [S.l.], v.8, n.3, p.279–286, Setembro 2004.

FONSECA, R.; SIMÕES, A. **Alternativas ao XML**: YAML e JSON. Disponível em: < http://alfarrabio.di.uminho.pt/ albie/publications/xmlyamljson07.pdf >. Acesso em agosto de 2010.

GOULARTE, R. **Personalisação e Adaptação de Conteúdo Baseadas em Contexto para TV Interativa**. 2003. 262p. Tese (Doutorado em Ciências Matemáticas e de Computação) — Universidade de São Paulo, USP, São Carlos, SP.

GREGORY, A.; ELIZABETH, M. Charting Past, Present, and Future Research in Ubiquitous Computing. **ACM Transactions onComputer-Human Interaction**, [S.l.], 2000.

GRIMM, R.; BERSHAD, B. . **Future Directions**: System Support for Pervasive Applications. Disponível em: < http://cs.nyu.edu/rgrimm/one.world/papers/fudico02.pdf/>. Acesso em novembro de 2009.

INFOEXAME. Smartphones, porque é hora de comprar um e aposentar seu celular. , [S.l.], n.257, Agosto 2007.

ISAM. **InfraEstrutura de Suporte às Aplicações Móveis**. Disponível em: < http://www.inf.ufrgs.br/ isam/index.html>. Acesso em março de 2010.

JANSEN, E. e. a. A Programming Model for Pervasive Spaces. **In International Conference on Service-Oriented Computing**, Netherlands, Dezembro 2005.

JAVA. **About JAVA**. Disponível em: < http://www.java.com/ >. Acesso em outubro de 2010.

JENA. **Jena - A Semantic Web Framework for Java**. Disponível em: < http://jena.sourceforge.net/ >. Acesso em fevereiro de 2010.

JESUS, R. M. de. **MATIW 1451 - Monitoramento e Acionamento de Transdutores Inteligentes através da Web (Padrão IEEE 1451)**. 2007. Tese de Mestrado em Engenharia Elétrica — Faculdade de Engenharia/UNESP, Ilha Solteira, SP.

JOHNSON, T. **Uma Arquitetura de Computação Pervasiva para Trabalho de Campo**. 2005. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Pernambuco, UFPE, Recife, PE.

JSF. **JSF - JavaServer Faces**. Disponível em: < http://pt.wikipedia.org/wiki/JavaServer_Faces >. Acesso em outubro de 2010.

LNCC. **Laboratório Nacional de Computação Científica**. Disponível em: < http://www.lncc.br/eventoSeminario/eventoconsultar.php-idtevento=743 >. Acesso em abril de 2010.

LOPES, J. L.; PILLA, M. L.; YAMIN, A. C. EXEHDA: a Middleware for Complex, Heterogeneous and Distributed Applications. **Iberian-American Conference on Technology Innovation and Strategic Areas**, Rio de Janeiro, Brazil, Maio 2007.

MACIEL, R. S. P.; ASSIS, S. R. Middleware: Uma Solução para o Desenvolvimento de Aplicações Distribuídas. **Científico - Ano IV**, [S.l.], Janeiro 2004.

MEDEIROS, R. A. **Estudo de Três Casos de Telemedicina no Brasil, nos Períodos de 2005 e 2006**. 2009. 171p. Tese (Doutorado em Ciências) — Universidade de São Paulo, USP, São Paulo, SP.

MITCHELL, S. e. a. Context-Aware Multimedia Computing in the Intelligent Hospital. **In Proceedings of the 9th ACM SIGOPS European Workshop**, ACM Press, p.13–18, 2000.

PEDERSEN, J. B.; MOGENSEN, M.; BARDRAM, J. E. The ABC adaptive fusion architecture. **Proceedings of the 4th international workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006)**, [S.l.], v.182, 2006.

PERTMED, . **PERTMED - Sistema de TeleMedicina Móvel**. Disponível em: < http://pertmed.wkit.com.br/pertmed/doku.php >. Acesso em novembro de 2009.

PHAM, T. L.; SCHNEIDER, G.; GOOSE, S.; PIZANO, A. Composite Device Computing Environment: A Framework for Situated Interaction Using Small Screen Devices. **Personal Ubiquitous Computing**, [S.l.], v.5, n.1, p.25–38, 2001.

POSTGRESQL. **About PostgreSQL**. Disponível em: < http://www.postgresql.org/about/ >. Acesso em novembro de 2010.

RANGANATHAN, A. e. a. Towards a Pervasive Computing Benchmark. **In 3rd International Conference on Pervasive Computing and Communications Workshops (PerCom)**, [S.l.], Dezembro 2005.

RIBEIRO, A. G. C. D. **Sistema de Prontuário Médico com Monitoramento de Pacientes via GPRS**. 2009. Monografia de Engenharia de Computação — CT/UFRN, Natal, RN.

RIZZETTI, T. A. **Um Ambiente de Contexto Personalizado e Orietando a Tarefas na Arquitetura ClinicSpace**. 2010. Tese de Mestrado em Ciência da Computação — PPGI/UFSM, Santa Maria, RS.

RODRIGUEZ, M. D. e. a. Location-Aware Access to Hospital Information and Services. **IEEE Transactions on Information Technology in Biomedicine**, [S.l.], v.8, n.4, p.448–455, 2004.

SAHA, D.; MUKHERJEE, A. Pervasive Computing: A Paradigm for the 21st Century. **Computer, IEEE Computer Society Press**, [S.l.], v.36, n.3, p.25–31, 2003.

SANTOS, J. C. dos. **Sistema Eletrônico para Aquisição, Processamento e Armazenamento de Sinais Biológicos Baseados na Norma IEEE 1451.4**. 2006. Tese de Mestrado em Engenharia Elétrica — Escola Politécnica/USP, São Paulo, SP.

SATYANARAYANAN, M. Pervasive Computing: Vision and Challenges. **IEEE Personal Communications**, [S.l.], v.4, n.8, p.10–17, Agosto 2001.

SCHMIDT, D. Trends in distributed object computing. **Parallel and Distributed Computing Practices**, [S.l.], v.3, n.1, 2000.

SILVA, F. L. da. **ClinicSpace:** Modelagem de uma Ferramenta-Piloto para Definição de Tarefas Clínicas em um Ambiente de Computação Pervasiva Baseado em Tarefas e Direcionado ao Usuário-Final. 2010. Tese de Mestrado em Ciência da Computação — PPGI/UFSM, Santa Maria, RS.

STRUZIK, Z. R. e. a. Mobile Nurse: Platform for Ubiquitous Medicine. **Methods of Information in Medicine**, [S.l.], v.46, n.2, p.140–146, Dezembro 2007.

TANI, F. K. **Proposta de Desenvolvimento de Transdutores Inteligentes Baseados na Norma IEEE 1451 Aplicados a Redes Lonworks**. 2006. Tese de Mestrado em Engenharia de Computação — Escola Politécnica/USP, São Paulo, SP.

TENTORI, M.; FAVELA, J. Activity-Aware Computing for Healthcare. **IEEE Pervasive Computing**, [S.l.], v.7, n.2, p.51–57, Abril 2008.

TOMCAT APACHE, . **The Oficial Tomcat Apache Site**. Disponível em: < http://tomcat.apache.org>. Acesso em novembro de 2009.

VARSHNEY, U. Pervasive Healthcare. **IEEE Computer**, [S.l.], v.36, n.12, p.138–140, 2003.

VENECIAN, L. R. **EXEHDA-CS:** Um Mecanismo para Sensibilidade ao Contexto com Suporte Semântico. 2010. Tese de Mestrado em Ciência da Computação — PPGINF/CPOLI/UCPEL, Pelotas, RS.

VIEGAS, S. Telemedicine - Practicing in the Information Age. , [S.l.], p.13–18, 1998.

VOIDA, S. Integrating virtual and physical context to support knowledge workers. **IEEE Pervasive Computing**, [S.l.], v.1, n.3, 2002.

W3C. **About Web Service**. Disponível em: < http://www.w3.org/TR/ws-arch/whatis >. Acesso em agosto de 2010.

WARKEN, N. **EXEHDA-DA:** Uma Proposta de Controle da Adaptação Dinâmica ao Contexto na Computação Ubíqua. 2010. Tese de Mestrado em Ciência da Computação — PPGINF/CPOLI/UCPEL, Pelotas, RS.

WEGDAM, M. Awareness: A project on context aware mobile networks and services medical systems international. **In the Proceedings of the 14th Mobile and Wireless Communications Summit**, [S.l.], p.19–23, Junho 2005.

WEISER, M. The Computer for the 21 st Century. **Scientific American**, [S.l.], v.3, n.265, p.94–104, Setembro 1991.

WEISER, M.; GOLD, R.; BROWN, J. S. The Origins of Ubiquitous Computing Research at Parc in the Late 1980s. **IBM Syst. J.**, [S.l.], v.38, n.4, p.693–696, 1999.

WOOTTON, R.; CRAIG, J.; PATTERSON, V. (Ed.). **Introduction to Telemedicine**. Lodon: The Royal Society of Medicine Press Ltd, 2006. 206p. n.2.

XML. **About XML**. Disponível em: < http://www.xml.org/ >. Acesso em setembro de 2010.

YAMIN, A. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. 195p. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre, RS.

YAMIN, A. e. a. EXEHDA: adaptative middleware for building a pervasive grid environment. **In: Frontiers in Artificial intelligence and Applications - Self - Organization and Automatic Informatics**, New York, p.203–219, 2005.

YAU, S.; YU, W.; KARIM, F. Development of Situation-Aware Application Software for Ubiquitous Computing Environments. **Proceedings of the 26th Annual International Computer Software and Applications Conference - COMPSAC**, [S.l.], p.233–238, 2002.

ZYL, P. V.; KOURIE, D. G.; BOAKE, A. Comparing the Performance of Object Databases and ORM Tools. **ESPRESSO Research Group**, [S.l.], 2006.

ANEXO A PADRÕES DE SENSORES E ATUADORES

Como padrão para interconexão dos sensores e atuadores do uMED, está sendo avaliado a possibilidade de utilizar a família de normas IEEE 1451 - *Standards for Smart Transducer Interface for Sensors and Actuators*.

Com avanços das tecnologias de redes de computadores, e o crescimento da *Internet*, grandes empresas como a GM, Boeing, entre outras começaram a reavaliar como suas tecnologias de rede poderiam ser padronizadas através de tecnologias e protocolos associados a *Internet* (JESUS, 2007). Neste contexto, empregam-se os transdutores inteligentes (sensores e atuadores) em rede com a finalidade de transmitir, compartilhar e disponibilizar informações em tempo real, os quais são protagonistas atual no cenário da automação industrial. O termo transdutor deriva-se do latim *transducere*, que significa "conduzir através de". Um sensor executa uma ação de transdução e pode ser definido como sendo um dispositivo capaz de converter uma grandeza física ou química em um sinal elétrico. O atuador tem o principio inverso de um sensor, pois converte um sinal elétrico em uma grandeza física ou química. Portanto, um dispositivo sensor ou atuador é classificado como um transdutor (JESUS, 2007).

Como o mercado de transdutores é muito diversificado, os fabricantes de transdutores procuram maneiras de construí-los dotados de "inteligência", com capacidade de operar em redes e com o menor custo operacional possível (JESUS, 2007).

Atualmente diversas redes de controle de transdutores disponíveis são proprietárias. As redes proprietárias possuem baixa flexibilidade e restringem o cliente aos serviços de apenas um determinado fabricante, causando o aumento do custo operacional desse serviço. A conexão de transdutores inteligentes em redes de controle que suportem uma grande variedade de protocolos (redes abertas) exige esforços significativos, aumentando consideravelmente a complexidade do sistema de automação. Entretanto, o sistema torna-se mais eficiente e com grande possibilidade de reduzir os custos operacionais. O ideal seria termos os transdutores inteligentes operando em redes totalmente abertas e padronizadas. Com esse objetivo surge o padrão proposto pelo comitê do IEEE 1451 (JESUS, 2007).

A premissa primcipal do IEEE 1451 é desenvolver uma interface padronizada para conectar transdutores a qualquer tipo de rede de comunicação. Assim, a interface padronizada deve utilizar as tecnologias de rede de controle existentes e suportar os diferentes tipos de sensores e atuadores, tornando este dispositivo inteligente (JESUS, 2007).

É importante salientar que a IEEE 1451 não propõe uma nova rede de controle. A idéia é fornecer um conjunto de interfaces padronizadas que permita utilizar as tecnolo-

gias de rede de controle existentes, promovendo a interoperabilidade entre transdutores e as mesmas. Em última análise, a IEEE 1451 permite que os fabricantes de transdutores se concentrem no desenvolvimento de seus produtos sem se preocupar com sua adequação a uma rede específica. Os usuários, por sua vez, podem escolher a rede de controle mais adequada à sua solução sem se preocupar com a disponibilidade de transdutores adequados a tal rede. Através das interfaces padrão definidas pela IEEE 1451, os transdutores podem ser conectados a diversas redes de controle de maneira transparente. Da mesma forma, é possível que transdutores de diferentes fabricantes sejam substituídos entre si sem impacto na solução existente (TANI, 2006).

Uma analogia simplificada pode ser feita com a interface USB (*Universal Serial Bus*) utilizada em microcomputadores. Diversos dispositivos USB[1] (transdutores) podem ser conectados ao computador independentemente do sistema operacional utilizado (rede de controle), pois a interface é padronizada. O fabricante do dispositivo não precisa se preocupar com o sistema operacional que será utilizado. Da mesma forma, o usuário do dispositivo pode substituí-lo por um de outro fabricante sem se preocupar com sua adequação ao seu sistema. Outra característica importante da IEEE 1451 é promover o desenvolvimento de dispositivos plug-and-play[2] (TANI, 2006) .

A utilização da norma IEEE 1451 em aplicações na área biomédica estão se proliferando devido a facilidade de funcionamento, autocalibração e autoreconhecimento dos elementos sensores (SANTOS, 2006).

A possível escolha da família de normas IEEE 1415 na concepção do uMED se deve a estudos que associam o uso da IEEE 1451 a diversas áreas de aplicação, dentre elas a biomédica, automação agrícola, indústria de alimentos, setor automobilístico e também o padrão IEEE 1451.5 merece destaque devido à popularidade e ao potencial de aplicação de tecnologia sem fio nos mais diversos campos (SANTOS, 2006; JESUS, 2007).

[1]Existem, no entanto, dispositivos USB que não são inteiramente compatíveis com outros sistema por razões diversas, como a indisponibilidade drivers específicos.

[2]Plug-and-Play: Capacidade de um dispositivo computacional ser identificado automaticamente e configurado dinamicamente.

ANEXO B CARACTERIZANDO SINAIS VITAIS DE PACIENTES

Sinais vitais, ou sinais cardinais, são aqueles que evidenciam as alterações da função corporal. Quando se desviam do normal, significa que o paciente precisa ser observado para se avaliar à relação causa e efeito (RIBEIRO, 2009).

As verificações mais frequentemente realizadas pelos profissionais de saúde são aquelas relativas à temperatura corporal, pressão arterial, pulso, frequência respiratória e saturação do oxigênio. Enquanto indicadores do estado de saúde, estas medidas revelam a eficácia das funções corporais circulatória, respiratória, neural e endócrina. Devido a sua importância, elas são denominadas sinais vitais (RIBEIRO, 2009).

Temperatura Corporal

De acordo com Ribeiro (RIBEIRO, 2009) , temperatura corporal é o grau de calor que o corpo apresenta. É o equilíbrio entre o calor produzido e eliminado pelo corpo. Há uma variação precisa de temperatura dentro das quais as células funcionam com eficiência e a atividade enzimática[1] é adequada.

A faixa normal de temperatura para adultos considerados normais é de 36°C a 37°C, a mesma é sucetível de pequenas variações de acordo com o local do corpo em que for tomada, por exemplo:

- Axilar (36°C a 37°C);
- Inguinal (36°C a 36,8°C);
- Bucal (36,8°C a 37°C);
- Retal (37°C a 37,2°C).

Para o seu conforto, o ser humano necessita manter sua temperatura central[2] dentro de uma faixa térmica estreita, por isso é dito que o homem é um animal homeotérmico. Se a temperatura corporal tornar-se se excessivamente alta ou baixa, todos os sistemas do organismo serão afetados, resultando na morte das células e consequentemente de todo sistema.

[1] Atividade enzimática é a capacidade das células em produzir enzimas benéficas ao organismo.

[2] Temperatura central é a temperatura do sangue arterial nas regiões centrais do corpo.

Pressão Arterial

A pressão ou tensão arterial é um parâmetro de suma importância na investigação diagnóstica, sendo usualmente praticada em consultas de diversas especialidades, relacionando-se com o coração, traduz o sistema de pressão vigente na árvore arterial. É medida na maioria das vezes com a utilização do esfigmomanômetro e do estetoscópio.

A pressão arterial no sistema de artérias no corpo, é um excelente indicador da saúde cardiovascular. Os níveis de pressão arterial normal variam ao longo da vida. Em uma criança ou adolescente é avaliado o tamanho corporal e a idade. Em um adulto, tende a aumentar com a idade avançada, o padrão para um adulto de meia-idade sadio é de 120/80 MmHg (Milímetro de Mercúrio). Na tabela B.1 é detalhado a pressão arterial normal conforme a faixa etária (RIBEIRO, 2009) .

Tabela B.1: Média de Pressão Arterial Normal (RIBEIRO, 2009)

Idade	Pressão Arterial
Recém nascido	40 (média)
1 mês	85/54
1 ano	95/65
6 ano	105/65
10-13 anos	110/65
14-17anos	120/75
Adulto médio	120/80
Idoso	140/90

Pulso (ou Frequência Cardíaca)

É o nome que se dá à dilatação pequena e sensível das artérias, produzida pela corrente circulatória. Toda vez que o sangue é lançado do ventrículo esquerdo para a aorta, a pressão e o volume provocam oscilações ritmadas em toda a extensão da parede arterial, evidenciadas quando se comprime moderadamente a artéria contra uma estrutura dura.

O pulso pode ser palpado em qualquer artéria, mas as mais fáceis de identificar são a radial no punho e a carótida[3] no pescoço (RIBEIRO, 2009).

A localização apical (lado esquerdo do peito) e radial são as comuns para exame da frequência cardíaca, porque mesmo que o débito cardíaco diminua, o sangue continua irrigando o cérebro, e os pulsos periféricos enfraquecem. Neste caso, a artéria carótida continua com frequência normal.

A frequência cardíaca é avaliada de acordo com fatores físicos, idade, estado psicológico e postura.

Segundo Ribeiro (RIBEIRO, 2009), a frequência cardíaca ou número de pulsação por minuto:

- Homem (60 a 70);

[3]Carótida: são vasos sanguíneos que levam sangue arterial ao cérebro. Cada ser humano tem duas artérias carótidas, começam no tórax passando pelo pescoço, uma de cada lado, chegando ao crânio.

- Mulher (65 a 80);
- Criança (120 a 125);
- Lactantes[4] (125 a 130).

Frequência Respiratória

Respiração é a troca de gases entre o organismo e o meio exterior; consistem na absorção de O2 (Oxigênio) e eliminação de CO2 (Gás Carbônico). A frequência normal de respiração no adulto varia de 16 a 20 vezes por minuto. Em crianças, de 20 a 25 por minuto, em lactantes de 30 a 40 por minuto (RIBEIRO, 2009).

Saturação do Oxigênio

Medir a saturação de oxigênio no sangue periodicamente é muito útil em diversos casos, principalmente em pacientes com obstrução pulmonar crônica ou doenças respiratórias. Para medir a saturação de oxigênio utiliza-se um dispositivo denominado oxímetro de pulso (RIBEIRO, 2009).

O oxímetro do pulso mede continuamente e de maneira não invasiva a saturação do oxigênio da hemoglobina arteriolar num local de medição periférica (i.e. pé, dedo do pé, ou dedo da mão). É utilizado para monitorizar doentes em risco de desenvolver hipoxemia[5]. O monitoramento da SpO2 (Saturação do Oxigênio) fornece informação acerca dos sistemas cardíaco e respiratório e dados sobre o transporte do oxigênio no organismo. É amplamente utilizado por ser não-invasivo e monitorar de maneira contínua, além de ser facilmente aplicado, sem dor.

O sensor do oxímetro contém dois LEDs (Díodo Emissor de Luz) que transmitem raios de luz vermelha e infravermelha através das extremidades do corpo. A luz transmitida é então recebida por fotodetector. O sangue saturado de oxigênio absorve a luz de uma maneira diferente do sangue não saturado de oxigênio. Assim, a quantidade de raios vermelhos e infravermelhos absorvida pelo sangue que flui numa área periférica adequada do corpo, normalmente o dedo em adultos e o pé em neonato, pode ser utilizada para calcular a relação entre hemoglobina oxigenada e hemoglobina total no sangue arterial. O monitor mostra esta relação como a percentagem de SpO2. Os valores normais situam-se, normalmente, entre 95 a 100 %, em localidades ao nível do mar (RIBEIRO, 2009).

[4]Lactantes: quem está amamentando ou produzindo leite.

[5]Hipoxemia: é a deficiência anormal de concentração de oxigênio no sangue arterial.