



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

***UbiDoctor: Arquitetura de Serviços para
Gerenciamento de Sessão e Adaptação de
Conteúdo em Ambientes de Medicina
Ubíqua***

Juliana Regueira Basto Diniz

Tese de Doutorado

Recife
26 de fevereiro de 2009

Universidade Federal de Pernambuco
Centro de Informática

Juliana Regueira Basto Diniz

UbiDoctor: Arquitetura de Serviços para Gerenciamento de Sessão e Adaptação de Conteúdo em Ambientes de Medicina Ubíqua

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: *Carlos André Guimarães Ferraz*

Recife
26 de fevereiro de 2009

Diniz, Juliana Regueira Basto

UbiDoctor: arquitetura de serviços para gerenciamento de sessão e adaptação de conteúdo em ambientes de medicina ubíqua / Juliana Regueira Basto Diniz – Recife : O Autor, 2009.

xxi, 156 folhas : il., fig., tab.

Tese (doutorado) – Universidade Federal de Pernambuco. Cln. Ciência da Computação, 2009.

Inclui bibliografia e apêndices.

1. Sistemas distribuídos. I. Título.

004.36

CDD (22.ed.)

MEI2009-019

Tese de Doutorado apresentada por **Juliana Regueira Bastos Diniz** a Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Ubidocor: Arquitetura de Serviços para Gerenciamento de Sessão e Adaptação de Conteúdo em Ambientes de Medicina Ubíqua**” orientada pelo **Prof. Carlos André Guimarães Ferraz** e aprovada pela Banca Examinadora formada pelos professores:



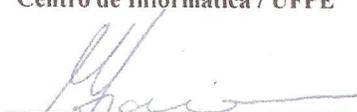
Prof. Nelson Souto Rosa
Centro de Informática / UFPE



Profa. Patricia Cabral de Azevedo Restelli Tedesco
Centro de Informática / UFPE



Prof. Geber Lisboa Ramalho
Centro de Informática / UFPE

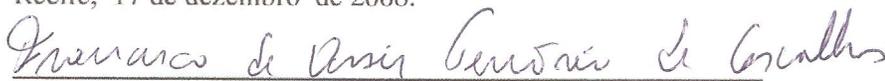


Profa. Magdala de Araújo Novaes
Núcleo de Telesaúde / UFPE



Profa. Thais Vasconcelos Batista
Departamento de Informática e Matemática Aplicada / UFRN

Visto e permitida a impressão.
Recife, 17 de dezembro de 2008.



Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

A minha filha Luíza

Agradecimentos

A concretização deste trabalho levou alguns anos e contou com o apoio e incentivo de várias pessoas que direta ou indiretamente apoiaram a sua realização. Em particular, agradeço:

A Deus, pois sem a Vossa vontade, nada seria possível.

Aos meus pais, Eliane e José Ricardo, por toda a educação que me proporcionaram durante toda a vida acadêmica, ensinando que a melhor forma de se construir algo é através do conhecimento, do empenho e dedicação.

A minha filha Luíza, que mesmo sem entender o porquê de minhas ausências no dia-a-dia, me escutava atentamente quando a explicava da necessidade de estudar.

Ao meu marido, Homero, que escutou por tantas vezes as minhas angustias e incertezas, me apoiando a cada fase do trabalho, trocando idéias e emitindo opiniões.

Ao meu orientador, Carlos Ferraz, pelo suporte, amizade e paciência ao longo deste trabalho.

À equipe da UCMF do Real Hospital Português, em especial, a Dra. Sandra Mattos por todo incentivo e parceria antes e durante a realização deste trabalho.

Aos colegas de caminhada Fernando Trinta e Sônia França com quem compartilhei minhas angustias, recebi sugestões e troquei idéias, ao longo de todo esse tempo de trabalho e aos colegas Hildeberto Melo e Leonardo Melo que colaboraram na implementação do protótipo e acreditaram na realização da pesquisa.

Ao colega Flávio Cireno, da Fundação Joaquim Nabuco, através dos conhecimentos repassados na área estatística e todo apoio prestado na avaliação do trabalho.

Às professoras Marizete Silva e Patrícia Tedesco pelo apoio, incentivo e colaboração para a concretização do trabalho.

Aos meus amigos e familiares que muitas vezes compartilharam minhas angustias, sempre com uma palavra de conforto e compreensão e em especial, a minha amiga Naide que escutou por várias vezes as minhas lamúrias e se faz tão presente em todas as fases de minha vida.

*Quanto mais abstrata for a verdade que queres ensinar,
mais tens que seduzir os sentimentos a seu favor.*

— FRIEDRICH NIETZSCHE

Resumo

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, em qualquer situação e a troca de opiniões e diagnósticos disponíveis no Prontuário Eletrônico do Paciente (PEP), permitindo que dados sobre exames, fatos e situações sobre a saúde de um paciente possam ser acessados através de múltiplos dispositivos e redes heterogêneas. Em particular, ambientes de medicina ubíqua precisam oferecer suporte à mobilidade de seus profissionais, uma vez que esta é uma característica inerente à própria profissão, especialmente dos médicos. Além do caráter nômade do médico, é importante considerar que a atividade médica está sujeita a interrupções durante sua execução, uma vez que médicos passam pouco tempo em cada local ou atividade. Essa fragmentação e nomadismo das atividades poderá comprometer a produtividade do médico. Dessa forma, mecanismos que facilitem a continuidade de atividades dos profissionais, mesmo em virtude de seus constantes deslocamentos, tendem a melhorar a sua produtividade. Este é o caso desta tese de doutorado que apresenta uma arquitetura baseada em serviços de *middleware*, denominada *UbiDoctor*, com objetivo de oferecer suporte a aplicações de medicina ubíqua, possibilitando motivar o médico a ficar menos ocioso entre atividades fragmentadas, podendo melhorar a sua produtividade. O suporte oferecido pelo *UbiDoctor* é dado pelos (i) serviços de gerenciamento de sessão, (ii) gerenciamento de contexto e (iii) adaptação de conteúdo. Estes serviços permitem que uma sessão iniciada em um dispositivo, possa ser concluída em outro, sem perda de informações e de tempo. O tempo de migração de sessões entre dispositivos foi verificado através de análises quantitativas e qualitativas com um grupo de médicos de um hospital particular em Recife-PE. Os resultados dos testes permitiram concluir que a solução computacional proposta possibilita o médico usar o sistema de PEP em qualquer lugar, a qualquer momento e de qualquer dispositivo, tendendo a melhorar sua produtividade.

Palavras-chave: Ubiquidade, Adaptação de Conteúdo, *Middleware*, Migração de Aplicações, Prontuário Eletrônico, Segunda Opinião Médica

Abstract

Ubiquitous healthcare environments include technological facilities that offer to their users new possibilities of access and interaction, such as the access to patients' data by using a portable device. In Ubiquitous medicine, the Electronic Patient Record (EPR) includes a set of information about a patient that can be accessed anywhere, anytime and using any device. These environments need to support the healthcare professionals' mobility, since this feature is an inherent characteristic of the physicians' job. It is also important to consider that a physician activity is frequently interrupted, since they do not spend so much time in the same place, performing a unique activity. Therefore, mechanisms that help the continuity of the healthcare professional's activity tend to improve their productivity. Thus, this thesis presents UbiDoctor, a service-based middleware infrastructure aiming at supporting ubiquitous healthcare applications provided by three services: (i) session management, (ii) context management and (iii) content adaptation. These services allow that a user starts a session in a specific device and finishes his work in another one, without losing his data and saving his time. UbiDoctor services were also evaluated, according to specific quantitative and qualitative metrics, by a group of physicians from a private hospital at Recife, Pernambuco. The results presented in this thesis show that Ubidoctor may improve a physician's productivity, by offering him ubiquitous access to an EPR application, almost anytime, anywhere and via heterogeneous devices.

Keywords: Ubiquity, Content Adaptation, Middleware, Application Roaming, EPR, Second Opinion

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Escopo	3
1.3	Requisitos do Cenário	5
1.4	O Cenário Proposto	6
1.4.1	Linha de Tempo do Cenário	8
1.5	Desafios Encontrados	10
1.6	Objetivos da Tese	12
1.7	Estrutura do Documento	13
2	Referencial Teórico	15
2.1	Conceito de Computação Ubíqua	15
2.2	Desafios da Computação Ubíqua	17
2.2.1	Heterogeneidade	17
2.2.2	Manutenção de Sessões	18
2.2.3	Adaptação de Conteúdo	19
2.2.3.1	Migração de Sessões e Adaptação de Conteúdo	21
2.2.4	Sensibilidade ao Contexto	22
2.2.4.1	Sistemas Sensíveis ao Contexto	23
2.2.4.2	Classificações de Informações de Contexto	25
2.3	Middleware com Suporte a Computação Ubíqua	26
2.4	Considerações Finais	29
3	Estado da Arte	31
3.1	<i>Middleware</i> e Medicina Ubíqua	32
3.2	<i>Middleware</i> e Sensibilidade ao Contexto	32
3.3	Medicina Ubíqua e Sensibilidade ao Contexto	38
3.4	Trabalhos Relacionados	39
3.4.1	Projeto ABC	39

3.4.2	Projeto Awareness	42
3.4.3	Projeto Marks	43
3.5	Considerações Finais	44
4	Arquitetura de Serviços <i>UbiDoctor</i>	47
4.1	Serviço de Gerenciamento Contexto	48
4.1.1	Componentes do Serviço de Gerenciamento de Contexto	51
4.1.2	Contexto de Localização	53
4.1.3	Contexto de Status	56
4.1.4	Contexto de Dispositivo	56
4.2	Serviço de Gerenciamento de Sessão	58
4.2.1	Políticas de Atualizações de Dados das Sessões	62
4.2.2	Atividades do Serviço de Gerenciamento de Sessão	67
4.2.3	Migração de Aplicações	70
4.2.3.1	Migração <i>Hardware-driven</i>	71
4.2.3.2	Migração <i>User-driven</i>	73
4.3	Serviço de Adaptação de Conteúdo	73
4.3.1	Políticas de Adaptação	76
4.3.2	Migração e Adaptação de Conteúdo	78
4.4	Considerações Finais	79
5	Prototipação	81
5.1	Arquitetura do Protótipo	81
5.2	Implementação dos Serviços	83
5.2.1	Serviço de Gerenciamento de Contexto	84
5.2.2	Serviço de Gerenciamento de Sessão	85
5.2.3	Serviço de Adaptação de Conteúdo	87
5.3	Implementação do Servidor de Aplicação	90
5.4	Implementação do Cliente	90
5.4.1	Interface Cliente para Web-Browsers	94
5.4.2	Interface Cliente para Telefones Celulares	97
5.5	Considerações Finais	101
6	Avaliação	103
6.1	Avaliação de Desempenho	103
6.1.1	<i>Desktop</i> para <i>Desktop</i>	105

6.1.2	Celular para <i>Desktop</i>	109
6.1.3	<i>Desktop</i> para Celular	112
6.1.4	Conclusões Obtidas com os Experimentos	115
6.1.5	Limitações Observadas nos Experimentos	117
6.2	Testes com Usuários	118
6.2.1	Metodologia Adotada	119
6.2.2	Perfil dos Participantes	119
6.2.3	Resultados Obtidos	120
6.2.4	Avaliação de Funcionalidades	122
6.2.5	Principais Conclusões dos Experimentos com Usuários	124
6.2.6	Limitações	125
6.3	Considerações Finais	125
7	Conclusão	127
7.1	Principais Contribuições	129
7.2	Trabalhos Futuros	130
7.3	Publicações do Trabalho Realizado	131
A	HISTOGRAMAS	143
B	TERMO DE COMPROMISSO	151
C	QUESTIONÁRIO DE AVALIAÇÃO	153
C.1	Identificação do Usuário	153
C.2	Uso de PEPs	153
C.3	Sobre os Serviços	154

Lista de Figuras

1.1	Arquitetura do Middleware	4
1.2	Cenário do <i>UHS</i>	7
1.3	Linha do Tempo	9
2.1	Computação Ubíqua (Saha e Mukherjee 2003)	16
2.2	<i>Framework</i> para Computação Ubíqua (Saha e Mukherjee 2003)	17
2.3	Camadas do Contexto (Neto 2006)	24
3.1	Áreas Temáticas	31
3.2	Arquitetura do CodeBlue (Lorincz et al. 2004)	33
3.3	Arquitetura do Infracore (Calv C. Z.; Pessoa e Filho 2005)	35
3.4	Visão Geral da Arquitetura do MoCA (Sacramento et al. 2004)	36
3.5	Modelo Lógico de Componentes (Bardram e Christensen 2001)	41
3.6	Camadas do Awareness (Wegdam 2005)	42
3.7	Arquitetura do Marks (Sharmin e Ahmed 2006)	43
4.1	Arquitetura de Serviços do <i>UbiDoctor</i>	48
4.2	Exemplo de Mudança de Estado	50
4.3	Esquema gráfico do Serviço de Contexto	51
4.4	Componentes do Serviço de Contexto	52
4.5	Mudança do Contexto de Localização - Saída de Hospital	55
4.6	Diagrama de Mudança de Contexto de Dispositivo Hardware-driven	58
4.7	Diagrama Mudança de Contexto de Dispositivo User-driven	59
4.8	Componentes do Serviço de Gerenciamento de Sessão	60
4.9	Diagrama de Classes para Informações de Sessão	61
4.10	Cenário para Sessão Composta	64
4.11	Cenário com Retorno após o Término da Sessão	65
4.12	Cenário Sessão Simples	66
4.13	Caso de Uso do Gerenciamento de Sessão	67
4.14	Diagrama de Colaboração - Entrada no Sistema	68

4.15	Cenário Sessão Simples com Migração	72
4.16	Diagrama de Seqüência - Adaptação de Conteúdo	74
4.17	Diagrama de Classes do Serviço de Adaptação de Conteúdo	75
4.18	Aplicação das Estratégias de Adaptação	78
4.19	Diagrama de Seqüência - Migração de Sessão <i>Hardware-driven</i> e Adaptação	79
5.1	Arquitetura do Protótipo	83
5.2	Pacote <code>services.device</code>	84
5.3	Inicialização do Serviço de Gerenciamento de Contexto	85
5.4	Interface do Serviço de Gerenciamento de Sessão	86
5.5	Inicialização do Serviço de Gerenciamento de Sessão	87
5.6	Serviço de Adaptação de Conteúdo	88
5.7	Implementação da <code>StopList</code>	89
5.8	Entidades do Banco de Dados	91
5.9	Diagrama de Classes do Servidor de Aplicação	92
5.10	Tela Principal de acesso via Web Browser	93
5.11	Tela para Incluir Novo Paciente	95
5.12	Telas para Consultar um Paciente Existente e seus Casos	96
5.13	Telas para Consultar e Responder Pareceres	98
5.14	Tela Principal de acesso via Celulares	100
5.15	Tela de Casos e Pareceres via Celulares	100
6.1	Tempo de migração versus Número de sessões	106
6.2	Tempo de migração versus Número de sessões	110
6.3	Tempo de migração versus Número de sessões	113
6.4	Síntese dos 3 Experimentos	116
A.1	Histograma do Tempo de Migração - Tela Consulta para Experimento 1	144
A.2	Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 1	145
A.3	Histograma do Tempo de Migração - Tela Consulta para Experimento 2	146
A.4	Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 2	147
A.5	Histograma do Tempo de Migração - Tela Consulta para Experimento 3	148
A.6	Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 3	149
C.1	Situações onde a Migração poderia ser Útil	155

Lista de Tabelas

3.1	Síntese de Trabalhos Relacionados	45
4.1	Classes de Dispositivos	49
4.2	Status do Usuário	49
6.1	Teste de Qui-Quadrado para o Experimento 1 - Tela Consulta Dados	107
6.2	Teste de Qui-Quadrado para o Experimento 1 - Tela Responder Parecer	107
6.3	Teste das Diferenças entre Médias - Experimento 1	108
6.4	Teste das Diferenças entre Médias - Experimento 1	108
6.5	Regressão Linear para o Tempo de Migração - Experimento 1	108
6.6	Regressão Linear para o Tempo de Migração - Experimento 1	109
6.7	Teste de Qui-Quadrado para o Experimento 2 - Tela Consulta Dados	110
6.8	Teste de Qui-Quadrado para o Experimento 2 - Tela Responder Parecer	110
6.9	Teste das Diferenças entre Médias - Experimento 2	111
6.10	Teste das Diferenças entre Médias - Experimento 2	111
6.11	Regressão Linear para o Tempo de Migração - Experimento 2	112
6.12	Regressão Linear para o Tempo de Migração - Experimento 2	112
6.13	Teste de Qui-Quadrado para o Experimento 3 - Tela Consulta Dados	113
6.14	Teste de Qui-Quadrado para o Experimento 3 - Tela Responder Parecer	114
6.15	Teste das Diferenças entre Médias - Experimento 3	114
6.16	Teste das Diferenças entre Médias - Experimento 3	114
6.17	Regressão Linear para o Tempo de Migração - Experimento 3	115
6.18	Regressão Linear para o Tempo de Migração - Experimento 3	115
6.19	Síntese de Funcionalidades Essenciais	123
6.20	Síntese de Funcionalidades Desejáveis	123

CAPÍTULO 1

Introdução

1.1 Motivação

O uso de tecnologia na prestação de serviços de saúde vem sendo realidade nos últimos anos (Novaes et al. 2004), (Novaes e Barbosa 2001). As atividades de segunda opinião e telediagnóstico remoto são utilizadas mediante uma infra-estrutura de rede, realizadas por profissionais conectados em outro centro médico distante da localidade onde o paciente está sendo examinado. Estes cenários são 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 Prontuário Eletrônico do Paciente (PEP), e através do acesso aos dados é possível se obter um diagnóstico mais completo.

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).

A partir do desenvolvimento de técnicas digitais de tratamento de informações clínicas, a figura do PEP começou a se tornar uma realidade (Liu et al. 2001). O uso simultâneo e distribuído deste importante conjunto de informações por diversos profissionais é um atrativo adicional (Furuie et al. 2003).

Com a evolução das redes de comunicação, permitindo larguras de banda superiores, comunicação sem fio e acesso através de dispositivos de menores dimensões, como celulares, *tablet PC* e *Personal Digital Assistant (PDA)*, 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 M.;Gold e 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 diferentes dispositivos e tecnologias de comunicação.

Weiser e colegas (Weiser M.;Gold e Brown 1999) argumentam que computadores de pequenas dimensões 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 melhorar a execução de suas aplicações.

Em ambientes de prestação de serviços de saúde, o suporte à mobilidade é uma questão importante, 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 (PC)*, *notebooks*, *tablets*, *palmtops*, *smartphones*, com configurações diversas e poderá fazer acesso ao sistema de informações de saúde de qualquer um deles. Também poderá ser permitido que o médico inicie a sessão usando um dispositivo, transfira-a para outro dispositivo, concluindo-a em um terceiro.

Para constatar o caráter nômade e fragmentado da atividade médica, observa-se o trabalho descrito por Trentoni e Favela (Tentori e Favela 2008). Eles realizaram um estudo de caso em um hospital público, onde observaram as práticas do corpo médico dentro do hospital. Verificou-se que os funcionários da equipe pesquisada (médicos, enfermeiros etc) 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 constatar 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 maior urgência.

Conforme pode ser observado, o cenário de aplicação desta tese é voltado para auxiliar médicos em sua rotina diária nos hospitais e consultórios, sobretudo no relacionamento com

outros médicos, no que diz respeito ao uso de PEPs, solicitação e respostas a segundas opiniões e juntas médicas. Considera-se um ambiente de computação ubíqua, aquele onde há diversos dispositivos computacionais (heterogêneos) nos vários locais por onde um médico passa, e o software de auxílio ao trabalho médico se adapta de forma sensível a contexto.

1.2 Escopo

As atuais soluções para este cenário consideram a mobilidade local (deslocamento do médico dentro de um hospital) e tratam de atividades que focam a relação médico-paciente numa rotina hospitalar. Para este trabalho de doutorado serão abordadas a mobilidade remota (deslocamento do médico do hospital para consultório externo ou residência, por exemplo) e a mobilidade local, conforme diferenciado por Bardram e colegas (Bardram J.; Kjær e C. 2003), focando-se nas ações de suspensão e migração de sessões, em tempo de execução, com possibilidade de adaptação de conteúdo contemplando a diversidade de dispositivos, preocupando-se com aspectos não funcionais como, por exemplo, tempo de migração e resgate de sessões, que ainda não foram tratados por outros trabalhos relacionados.

A mobilidade local envolve o deslocamento de pessoas entre cômodos internos a um ambiente (Bardram J.; Kjær e C. 2003). Bellotti e Bly (Bellotti e Bly 1996) argumentam que é necessário fazer a distinção entre mobilidade local e mobilidade remota, pois as necessidades de suporte a estes dois tipos de mobilidade são diferentes. A mobilidade remota, por sua vez, relaciona-se com o deslocamento do usuário entre prédios de localidades diferentes.

Na medicina ubíqua pode-se realizar acessos a sistemas de prontuários eletrônicos com informações consolidadas sobre os pacientes de qualquer lugar da rede, permitindo, inclusive, que haja cooperação entre profissionais independentemente do tempo e do espaço.

Operações de telediagnóstico, segunda opinião e junta médica são comuns em tais ambientes. Para viabilizar o acesso a partir de qualquer lugar, o médico poderá fazer uso de diversos dispositivos computacionais como, por exemplo, PC, *notebooks*, *tablets*, *palmops*, *smartphones* e de diferentes enlaces de comunicação (*Ethernet*, *WiFi*, *Bluetooth*, *3G*, etc) com diversos mecanismos de acesso (com fio, sem fio, rede de telefonia celular). Também é importante salientar que alguns dos dispositivos envolvidos apresentam limitações de recursos, como, por exemplo, restrições de memória, sistemas operacionais, tamanho de tela, teclado e bateria.

Para dar suporte ao cenário de medicina aqui proposto, pretende-se utilizar um conjunto de serviços de *middleware*. De acordo com Tanenbaum (Tanenbaum 2001), um *middleware* é uma camada de *software* que esconde a heterogeneidade do ambiente de execução, fornecendo

um modelo de programação para facilitar o desenvolvimento de aplicações distribuídas.

De acordo com Douglas Schmidt (Schmidt 2000), o *middleware* pode ser decomposto em três subcamadas conforme observado na Figura 1.1. São elas: infra-estrutura, distribuição e serviços comuns. A primeira subcamada encapsula os serviços de comunicação com o sistema operacional e serviços de concorrência minimizando os aspectos tediosos do desenvolvimento de aplicações distribuídas. Também possui a função de manter as aplicações usando mecanismos de baixo-nível da programação em rede, como *sockets*.

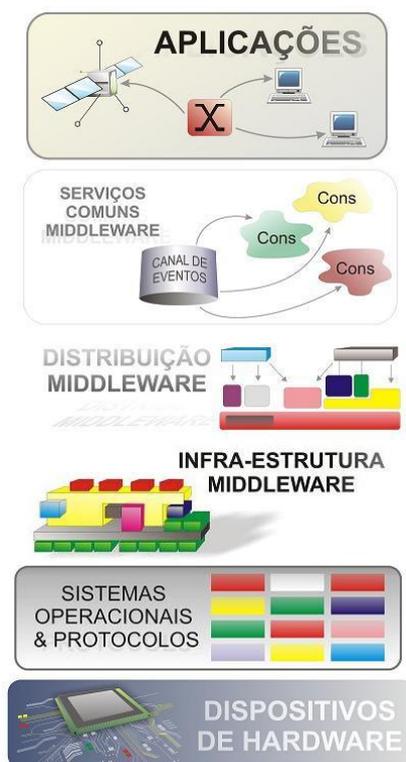


Figura 1.1 Arquitetura do Middleware

A subcamada de distribuição automatiza as tarefas da programação em rede tais como serialização e remontagem (*marshaling* e *unmarshaling*). Por fim, a subcamada de serviços comuns contém serviços reusáveis, independentemente do domínio. Essa subcamada inclui serviços de persistência, transações, tolerância a falhas, dentre outros. Os serviços propostos por este trabalho localizam-se nesta subcamada do *middleware*, pois podem ser aplicados a qualquer ambiente de mobilidade com características e requisitos similares.

Com o desenvolvimento de uma camada de serviços de *middleware* será possível fazer uso da transparência de distribuição, notificação de eventos, sensibilidade ao contexto, dentre

outros aspectos, através do uso de uma arquitetura, aqui denominada *UbiDoctor*, permitindo facilitar o desenvolvimento de aplicações de cenários complexos como o aqui descrito.

1.3 Requisitos do Cenário

No início do desenvolvimento desta pesquisa de doutorado, não estava claro que funcionalidades um *middleware* deveria ter para dar suporte a um ambiente médico ubíquo. Sendo assim, o primeiro passo foi realizar um levantamento de requisitos com os principais usuários de tais ambientes: os médicos. Para tal, foram realizadas reuniões com a equipe de cardiologistas do Real Hospital Português, em Pernambuco a fim de conhecer a rotina de trabalho vivenciada pelos médicos.

Inicialmente, foi realizada uma conversa informal com o líder da equipe de médicos que apresentou o panorama geral das necessidades dos médicos daquela equipe. Em seguida, outros médicos também foram consultados e algumas constatações foram observadas.

O uso de prontuários eletrônicos no hospital é bastante freqüente e vem aumentando a cada ano, de modo que novas informações vem sendo agregadas ao prontuário. De acordo com os relatos obtidos em conversas informais com o líder e demais médicos, a necessidade de ter acesso a dados do paciente a qualquer hora e local e usando algum dispositivo disponível é constante, já que a maioria dos médicos da equipe realizam atendimentos em consultórios interno e externo ao hospital, realizam exames em salas específicas no hospital e atuam na unidade de terapia intensiva (UTI) do hospital, através de plantões. Além disso, alguns médicos também atuam em outros hospitais da região metropolitana e possuem consultório particular.

Eles alegam que freqüentemente precisam ter acesso a informações clínicas de um determinado paciente e muitas vezes não se encontram em frente a um computador de consultório ou hospital. Essa necessidade se dá quando um colega solicita uma opinião a um determinado caso, ou quando o próprio paciente ou algum familiar consulta o médico para informar algum fato novo ou dúvida sobre medicamentos e procedimentos. Para os médicos, a possibilidade de se acessar prontuários eletrônicos de forma ubíqua, ajudaria a resolver esse problema, já que boa parte deles possuem, pelo menos, um telefone celular que o acompanha, onde quer que ele vá.

Os médicos mais acostumados a lidar com a tecnologia, além de telefones celulares, ainda possuem PDAs ou Tablet PCs. Entretanto, eles ainda não utilizam acesso a prontuários eletrônicos em tais dispositivos, pois não existe esta funcionalidade nos hospitais onde eles trabalham.

Os médicos também alegam que, não menos freqüentemente, são interrompidos, em meio

a suas atividade, por chamados emergenciais. Esses chamados são oriundos do mesmo hospital onde se encontram ou de outro hospital/consultório. Para realizar este atendimento emergencial, o médico precisaria interromper sua atividade e retorná-la algum tempo depois, precisando lembrar o que estava fazendo, verificar onde ele havia parado, para após retomar o contexto da situação, dar prosseguimento a atividade suspensa de outrora. Nesta ação, o médico perde tempo para comutar de uma atividade para outra, desprendendo um tempo com pouca ou até nenhuma utilidade.

Embora eles achem que o acesso ao PEP de dispositivos com tela pequena facilite a suas vidas, os mais familiarizados com esses dispositivos apresentam preocupações no sentido de como informações de grande volume como as encontradas em prontuários eletrônicos podem ser visualizadas em dispositivos de tela limitada.

Em síntese, os principais requisitos de um ambiente de medicina ubíqua para dar suporte a rotina de trabalho dos médicos consultados são:

- Suporte a fragmentação e mobilidade em suas atividades de rotina;
- Acesso ao ambiente de diversos pontos da cidade (residência, carro, hospitais e consultórios);
- Uso de diversos dispositivos, alguns com autonomia limitada;
- Acesso através de dispositivos com tela pequena.

1.4 O Cenário Proposto

Através do levantamento de requisitos, foi estabelecido um conjunto de situações que representem uma visão particular do ambiente de medicina ubíqua a ser atendido por esta trabalho. Este conjunto de situações constituem o cenário, denominado *Ubiquitous Health Service* (UHS), que é representado na Figura 1.2.

O UHS constitui uma rede de serviços de saúde, da qual fazem parte um conjunto de hospitais geograficamente distribuídos e médicos associados que podem ter acesso aos serviços desta rede de qualquer lugar, usando um conjunto de diferentes dispositivos e redes de acesso. Esta rede favorece o relacionamento entre os médicos associados, sendo portanto, o médico o usuário alvo do cenário. Os médicos, por sua vez, são colaboradores de alguns dos hospitais credenciados e podem acessar os serviços disponibilizados pelos hospitais, podendo fazê-lo também de sua residência, local de lazer, em trânsito ou do seu consultório.

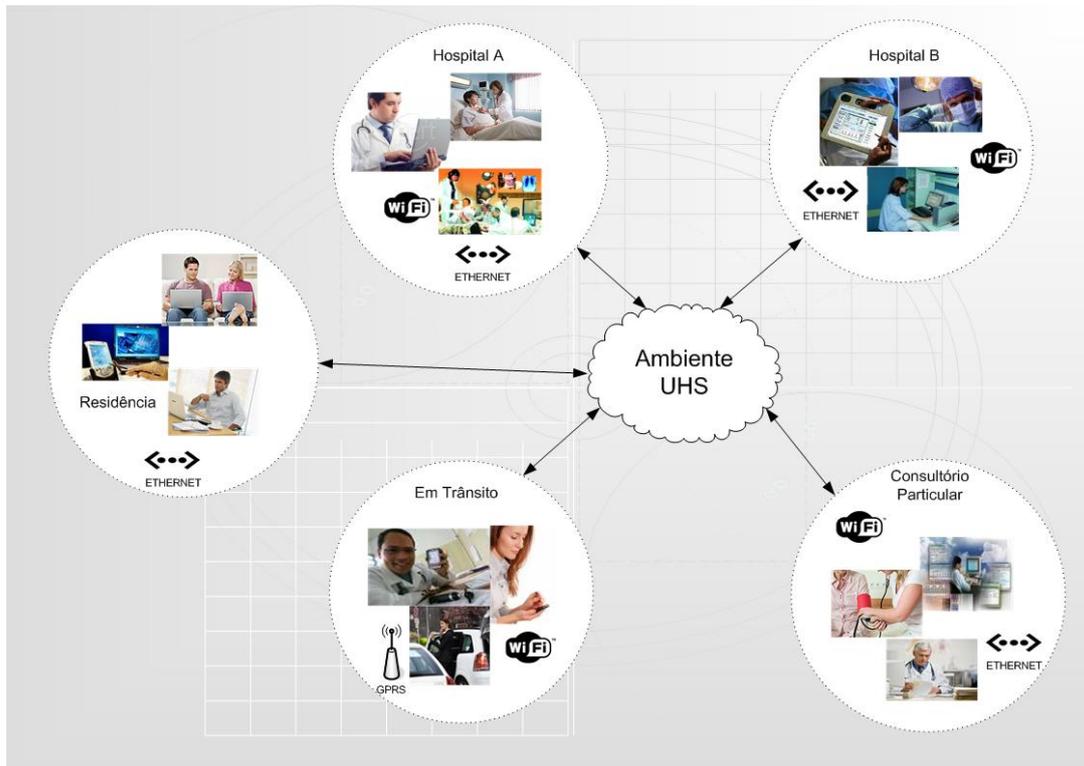


Figura 1.2 Cenário do UHS

Além de serviços específicos de cada hospital credenciado, também é possível ao médico realizar acesso ao PEP, a qualquer hora e de qualquer lugar, usando algum dispositivo disponível. Os hospitais, por sua vez, podem oferecer serviços específicos, como por exemplo, serviço de reserva de salas, marcação de consultas no ambulatório, localização de médicos, etc. Para estes serviços serão necessárias algumas validações de segurança e algoritmos de autenticação que são gerenciados pelo próprio hospital que disponibiliza os serviços.

O cenário possibilita ao médico iniciar uma sessão de acesso ao PEP usando um dispositivo, e transferi-la para outro dispositivo durante a sua execução. Uma sessão caracteriza-se por um intervalo de tempo em que um médico interage com um sistema de informações médicas a partir de um dispositivo específico, quer seja um dispositivo móvel ou um computador pessoal de mesa (desktop).

Em ambientes médicos como o UHS, a possibilidade de transferir sessões entre dispositivos é útil pelo fato da atividade médica ser bastante fragmentada, conforme citado na seção anterior. Embora a fragmentação na condução das atividades seja um fato concreto, a possibilidade do médico iniciar uma sessão em um dispositivo e retornar dentro de algum tempo usando o mesmo ou outro dispositivo apresenta uma possibilidade de aumento de produtividade em seu

trabalho, visto que ele leva um certo tempo para comutar entre as atividades executadas em sua rotina profissional (Tentori e Favela 2008).

Caso o tempo levado entre a migração e o retorno de uma sessão seja pequeno, de modo a representar um baixo impacto no tempo observado entre duas atividades executadas sucessivamente, o ambiente proposto possibilita que o seu trabalho possa ser mais produtivo, uma vez que permite incentivar o médico a assumir logo a sua próxima atividade.

Para melhor visualizar a rotina de um médico associado ao sistema, utiliza-se uma linha de tempo (*timeline*) onde são descritas uma série de situações nas quais o médico interage com o ambiente UHS. Essa rotina é descrita na Subseção 1.3.1.

1.4.1 Linha de Tempo do Cenário

Dra.Luíza acorda às 6h da manhã e prepara-se para a sua jornada de trabalho diária. Ela atende em seu consultório, pela manhã, mas, ao mesmo tempo, fica disponível para receber solicitações de colegas, com relação a casos clínicos interessantes, no estilo segunda opinião. Para isso, ela acessa o ambiente UHS, logando-se no PEP, através do seu telefone celular, habilitando o seu status como "disponível", pouco antes de sair de casa. No momento da sua entrada (*login*) no sistema, ela verifica se existem solicitações de pareceres pendentes, durante o período da noite, no qual não esteve disponível.

Dra.Luíza segue para seu consultório e durante o trajeto, seu celular recebe uma solicitação de parecer de um colega que se encontra em um hospital do outro lado da cidade. Seu colega é usuário da mesma rede de serviços e gostaria de obter uma segunda opinião sobre um caso clínico. Esta primeira seqüência de eventos pode ser observada no Quadro 1 da Figura 1.3.

Dra.Luíza inicia a análise do caso fazendo uso de exames e informações relacionadas através do seu celular porém, ao chegar ao seu consultório, ela prefere continuar a análise usando um dispositivo mais confortável (tela maior, por exemplo). Sendo assim, no momento em que chega à sua sala, Luíza solicita que o sistema indique outras possibilidades de dispositivos cadastrados e disponíveis naquela área em que se encontra.

O sistema, baseado no contexto dos dispositivos e do usuário, sugere algumas opções e ela escolhe o *desktop* da sua sala, solicitando, então a transferência da sessão para o mesmo, continuando a análise do caso clínico. Este processo de transferência da sessão recebe o nome de migração (Bardram 2004) (Sousa e Garlan 2002) de aplicações e pode ser interrompido por diversos fatores, fazendo com que a transferência da sessão não possa ser concluída e a sessão permaneça com o seu status pendente. Após concluir a sessão, Dra.Luíza inicia o atendimento das suas consultas, deixando seu status no sistema "ausente", o que significa que não estará

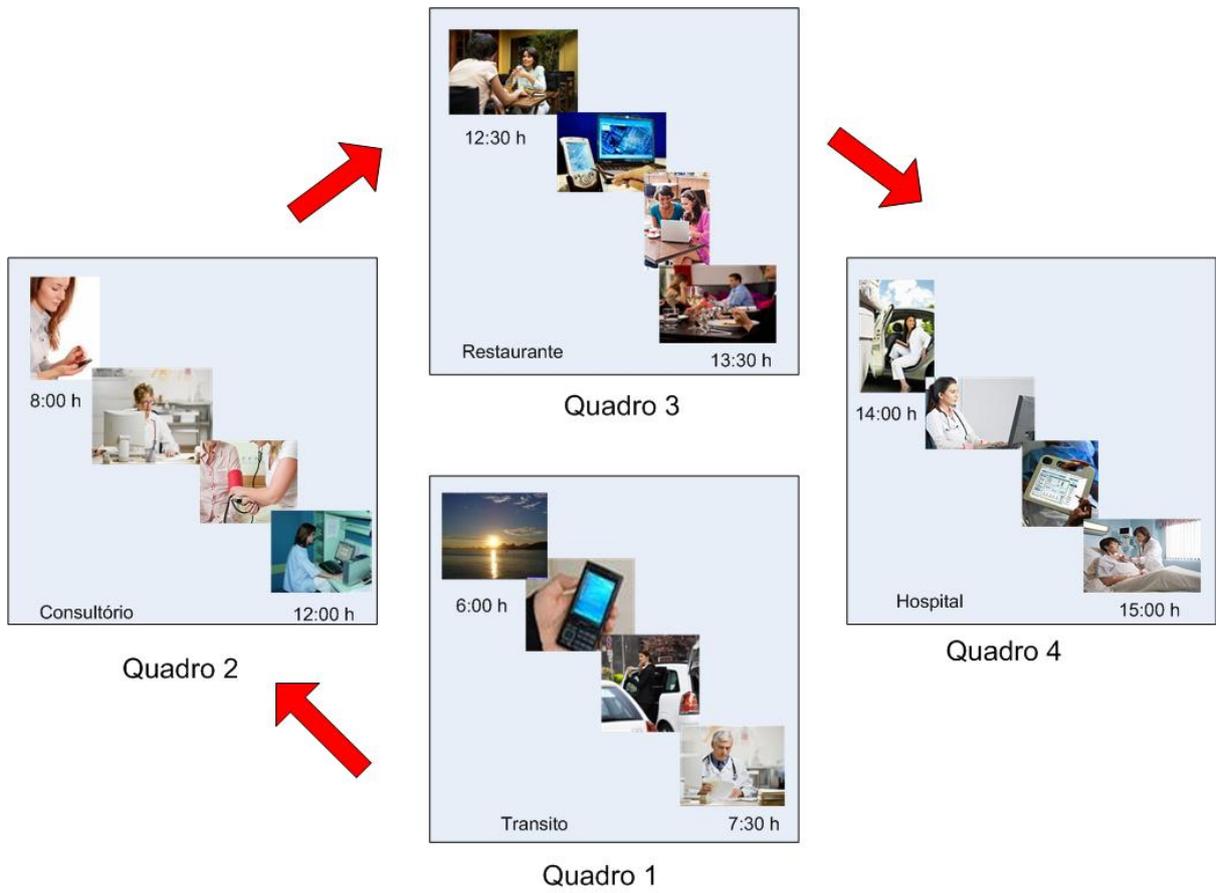


Figura 1.3 Linha do Tempo

próxima do dispositivo utilizado.

Posteriormente, ela precisará fazer uma marcação de um exame para um paciente em atendimento, a ser realizado em um determinado hospital credenciado. A marcação de exames é um serviço específico oferecido por este hospital que ela é colaboradora. Este segundo conjunto de situações está ilustrado no Quadro 2 da Figura 1.3.

Às 12 horas, ela termina as consultas e verifica no sistema se houve solicitações de pareceres. Como não houve, ela desconecta-se do sistema para o intervalo do almoço. Luíza vai almoçar em um restaurante com uma colega de trabalho. Durante o almoço elas conversam sobre um caso clínico de um paciente e precisam consultar alguns dados no PEP. Para isso, Luíza liga o seu PDA com acesso GPRS/EDGE/3G e acessa os dados do paciente em questão.

Durante o acesso aos dados, o PDA avisa que está com pouca bateria e solicita um novo dispositivo para migrar a sessão. Não há sugestões dadas pelo sistema, baseadas na localização, uma vez que elas se encontram em um local indeterminado (restaurante). Dessa forma, o sistema sugere os dispositivos pessoais de Luíza (celular, notebook), embora ela não esteja com nenhum deles. A colega sugere, então, usarem seu *notebook* e a sessão é transferida para o mesmo, antes até de elas fazerem o pedido do almoço ao garçon. Quando o almoço chega, elas ainda não haviam concluído a sessão, mas isto pode ser feito posteriormente, então Luíza suspende a sessão e vai almoçar. Essa seqüência de eventos pode ser observada no quadro 3 da Figura 1.3.

Ao chegar ao hospital que trabalhará no próximo turno (14-18 horas), ela acessa a sessão do PEP que havia sido suspensa no almoço, usando um *desktop*, e continua com a entrada de dados. Antes de concluir sua atividade no PEP, ela é interrompida por um chamado presencial de emergência, solicitando deslocamento para atendimento a um paciente em um leito da enfermaria. Como falta muito pouco para concluir a sessão, ela solicita a migração para o seu *tablet* e, enquanto desloca-se ao atendimento, conclui a sessão, conforme ilustrado no quadro 4 da Figura 1.3.

1.5 Desafios Encontrados

De acordo com o cenário descrito anteriormente, médicos precisam acessar PEPs, de qualquer lugar, com possibilidade de transferência de sessão entre dispositivos com diferentes configurações, em tempo satisfatório e com consistência de informações.

O ambiente UHS abrange uma grande diversidade de categorias de dispositivos móveis e fixos, ou seja, celulares, PDAs, *tablets*, *smart-phones*, *notebooks* e *desktops*. Cada um destes

dispositivos apresenta variações em tamanho de tela, sistema operacional, poder computacional e conexões de redes permitidas. Este **ambiente multiplataforma** pode ser considerado o primeiro desafio encontrado no cenário desta tese.

O segundo desafio diz respeito à **autonomia limitada de alguns dispositivos móveis**. Embora haja uma evolução das baterias, os dispositivos móveis que dependem de tal fonte de energia ainda possuem 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 concluí-la, já que o dispositivo utilizado não dispõe de energia suficiente para tal. Não se pretende tratar do problema das baterias propriamente dito, porém pretende-se encontrar uma solução de software que permita que o médico 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.

O terceiro desafio refere-se à possibilidade de **interrupção de uma sessão** de acesso a um PEP, inclusive, no momento em que a transferência de sessão entre dois dispositivos esteja acontecendo. A transferência de sessão poderá acontecer por uma necessidade, como por exemplo, o aviso de término da bateria de um dispositivo portátil ou por uma preferência de usuário. Em algumas situações, no ambiente UHS, num dispositivo com interface limitada, como um PDA ou celular, 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*.

É importante perceber que existem diversas possibilidades de tecnologias de redes envolvidas no cenário, a exemplo de *WiFi*, *bluetooth*, IEEE802.3, GSM, dentre outras. Essas diferenciações de tecnologias são ainda mais notáveis quando se trata de um ambiente de rede metropolitana, que perpassa os limites de um hospital ou uma clínica. O médico pode acessar o sistema em variadas localizações na 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. Assume-se que o *handoff* entre diferentes tecnologias de comunicação é possível, conforme proposto pelo trabalho de Cavalcanti e colegas (Cavalcanti et al. 2005). Essa diversidade de enlaces de rede possibilita a interrupção de sessões de forma involuntária, devido a problemas de **intermitência de comunicação**.

A tarefa de transferência de sessão entre dispositivos é uma tarefa complexa e deve acontecer, em muitos casos (foco desta tese), com o menor atraso possível, quando for solicitada. 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 importante 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, pois 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 empregada na área médica (Bardram 2005) e ambientes ubíquos em geral (Sousa e Garlan 2002). Como mencionado anteriormente, a migração de aplicação poderá acarretar problemas quando for interrompida de forma involuntária. Não foram encontrados trabalhos relacionados que tratem dessa interrupção de migração de sessão, nem do tempo em que ela ocorre. Estes são problemas que se deseja tratar nesta tese de doutorado.

No ambiente UHS, o tratamento a ser dado a estes desafios 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 o novo dispositivo de destino. Para que haja essa migração sem perda de informação e de tempo, o contexto do dispositivo destino tem que ser conhecido e controlado através de ações realizadas pelo serviço de gerenciamento de contexto. A abordagem utilizada para o tratamento da multiplataforma, sobretudo no que diz respeito a variações de tamanho de tela, é a adaptação de conteúdo.

Para superar os desafios mencionados, este trabalho investiga a hipótese de que se for definida 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 no cenário UHS, descrito anteriormente.

1.6 Objetivos da Tese

O objetivo geral desta tese de doutorado é definir uma arquitetura de serviços para gerenciamento de sessões que permita que uma sessão de uma aplicação (exemplo PEP) possa migrar de um dispositivo para outro, durante a sua execução, em tempo hábil e com informações essenciais preservadas.

Dentre os objetivos específicos deste trabalho de doutorado estão:

- Definir um serviço que gerencie as sessões de aplicações, controlando persistência, interrupções e migração de sessões;
- Agregar ações de adaptação de conteúdo baseadas em variáveis de contexto, tendo em vista a variedade de dispositivos envolvidos no cenário;

- Implementar e validar a arquitetura proposta, através de um protótipo na área médica.

1.7 Estrutura do Documento

Para alcançar os objetivos acima definidos, esta tese está dividida em seis capítulos. O presente capítulo apresenta a motivação, o cenário e os desafios observados no mesmo.

O Capítulo 2 apresenta o referencial teórico de computação ubíqua, *middleware* e sensibilidade ao contexto. Os trabalhos relacionados às áreas temáticas, envolvendo medicina ubíqua, sensibilidade ao contexto e *middleware* são apresentados no Capítulo 3.

O Capítulo 4 apresenta a definição da arquitetura de serviços (*UbiDoctor*) e o projeto e implementação dos seguintes componentes:

- serviço de gerenciamento de sessões;
- serviço de adaptação de conteúdo;
- serviço de gerenciamento de contexto.

O Capítulo 5 apresenta a implementação do protótipo de uma aplicação médica, em um cenário que utiliza PEP e faz uso da arquitetura de serviços apresentada. No Capítulo 6 será apresentada a avaliação da arquitetura proposta através do protótipo, utilizando os seguintes resultados:

- Análise da solução proposta, mostrando o tempo de migração versus o número de sessões simultâneas em diferentes configurações de redes e dispositivos;
- Obtenção das impressões do usuário (médico) com relação ao protótipo, através de uma pesquisa qualitativa;

Por fim, o Capítulo 7 apresenta as conclusões do trabalho, as suas contribuições esperadas e propostas de trabalhos futuros.

Referencial Teórico

O crescimento da computação ubíqua é fato notório nos últimos anos (Saha e Mukherjee 2003), (Grimm e Bershad 2002). As aplicações distribuídas devem acompanhar essa tendência, sendo capazes de serem executadas em ambientes ubíquos, na presença de uma diversidade de dispositivos.

Este capítulo aborda os conceitos principais e os desafios a serem vencidos por estudos desenvolvidos na área de computação ubíqua, bem como aspectos referentes a *middleware* para estes ambientes.

2.1 Conceito de Computação Ubíqua

De acordo com Mark Weiser (Weiser M.;Gold e Brown 1999), 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, *smart-phones* e a consolidação de padrões para redes sem fio como o *Bluetooth* (Bluetooth.org 2004) e o IEEE 802.11 (IEEE 2005). 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).

A computação ubíqua ganhou muito espaço com a disseminação de dispositivos portáteis, sobretudo após a ampliação das tecnologias de redes sem fio. Dessa forma, o uso da computação tem sido trazido para diversos cenários reais, onde sua usabilidade nos moldes da computação tradicional não poderia ser tão ampla. Cenários na área médica, podem extender-se a diversas atividades, como telemonitoramento através do uso de sensores de medicina ubíqua, diagnósticos remotos, segunda opinião, juntas médicas, viabilizando a integração de dispositivos móveis como PDAs, *tablets*, *smart-phones* com os sistemas de prontuário eletrônico existentes nos hospitais, dentre outros cenários, desde residências inteligentes a trabalhos colaborativos.

A idéia básica por trás do conceito de computação ubíqua é desenvolver uma variedade

de dispositivos inteligentes para serem utilizados em ambientes de trabalho e residencial. Tais dispositivos provêm aos usuários acesso universal e imediato às informações e dão suporte às tarefas dos usuários (Grimm e Bershad 2002).

O esquema da computação ubíqua é apresentado na Figura 2.1. Além de mobilidade, os sistemas ubíquos requerem suporte à interoperabilidade, escalabilidade, dentre outros aspectos para garantir que os usuários tenham acesso quando desejarem (Saha e Mukherjee 2003).

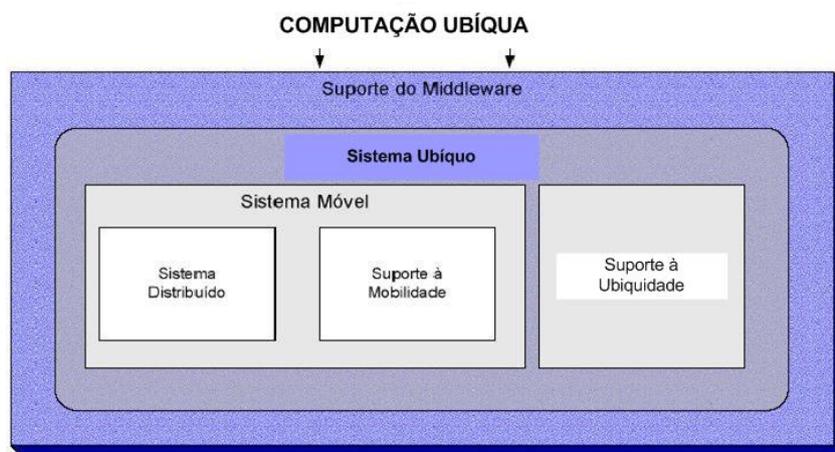


Figura 2.1 Computação Ubíqua (Saha e Mukherjee 2003)

Segundo Saha e Mukherjee (Saha e Mukherjee 2003), os avanços tecnológicos necessários para construir um ambiente de computação ubíqua são os seguintes: dispositivos, rede de interconexão, *middleware* e aplicações (Figura 2.2).

De acordo com 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 inteligentemente a informações do mesmo, acessível a qualquer hora e lugar. Para Couloris e colegas (Couloris G. ; Dollimore e Kindberg 2005), um espaço inteligente é qualquer espaço físico com serviços embarcados, ou seja, serviços providos sobretudo dentro daquele espaço físico.

Quando se refere a ambientes de medicina ubíqua, como por exemplo o UHS, o conceito de espaços inteligentes deverá possibilitar a realização das seguintes ações:

- Permitir que os profissionais de saúde, através de dispositivos portáteis ou não, possam ter acesso às informações do PEP, de qualquer local, desde que estejam conectados à rede. O ambiente se encarregará de gerir a comunicação de usuários e dispositivos às fontes de informações;

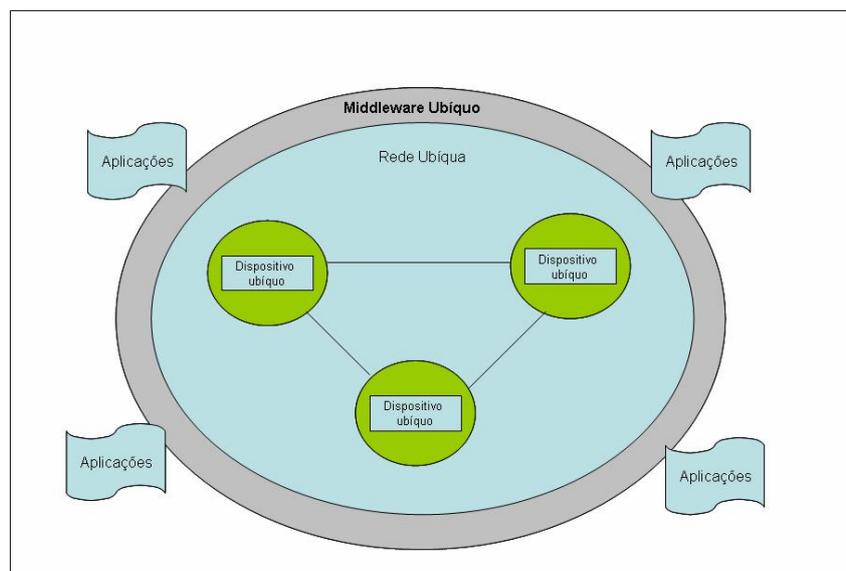


Figura 2.2 Framework para Computação Ubíqua (Saha e Mukherjee 2003)

- Habilitar a migração de aplicações de modo que uma aplicação sendo executada por um dispositivo possa ser transferida para outro dispositivo, dando continuidade a sessão;
- Gerenciar as sessões de PEPs de diferentes usuários, usando diferentes dispositivos e redes de acesso;
- Utilizar informações contextuais para auxiliar na disseminação da informação, automação de configuração e adaptação de conteúdo.

2.2 Desafios da Computação Ubíqua

Como citado previamente, 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, pois foram identificados no cenário UHS, e serão apontados nesta seção.

2.2.1 Heterogeneidade

A diversidade de dispositivos, configurações, modelos, arquiteturas de *software* e conexões de rede são aspectos presentes em ambientes ubíquos. A heterogeneidade em relação a *hardware* e *software* é um fato e ao mesmo tempo traz o desafio da necessidade de integração deste ambi-

ente multiplataforma. Acrescenta-se ainda que, quando o número e a variedade de dispositivos aumenta, torna-se maior a complexidade para tornar real a integração (Saha e Mukherjee 2003).

Um outro problema que atinge a questão da diversidade de dispositivos está associado às diferentes configurações que os mesmos podem possuir. Alguns, sobretudo os portáteis, apresentam restrições de processamento, comunicação e apresentação. A necessidade de serem leves e pequenos faz também com que suas baterias sejam pequenas e com pouca autonomia de vida. Todo tipo de processamento consome a energia das baterias destes dispositivos e, em particular, a comunicação sem fio é uma atividade que requer um alto consumo de energia.

2.2.2 Manutenção de Sessões

A computação ubíqua envolve troca de dados entre dispositivos móveis durante o acesso aos serviços. Existem diversas maneiras de um servidor manter certas informações requeridas no período de duração de uma sessão. Uma maneira de manter as informações relacionadas a uma sessão é associando-a a um processo (Lee et al. 2002). Entretanto, o servidor poderá apresentar degradação de desempenho devido à sobrecarga de processos criados para representar sessões num ambiente de grande abrangência.

Um mecanismo alternativo é a utilização de *Dynamic Link Library* (DLL). Como a DLL compartilha espaços em memória, poderá evitar a sobrecarga da mesma através da liberação de espaço. Contudo, este mecanismo gera problemas na manutenção das informações da sessão após a DLL ser desabilitada. Dessa forma, a informação da sessão deve ser armazenada em um meio de armazenamento persistente, como um banco de dados ou um sistema de arquivos. O trabalho referenciado em (Lee et al. 2002) apresenta um gerenciador de sessão que controla as informações temporárias associadas a uma sessão, durante a sua existência. O papel deste gerenciador é manter e controlar toda a informação de uma sessão desde a sua inicialização até o seu término.

Lee e colegas implementaram 3 modelos de gerenciadores de sessão (Lee et al. 2002). O primeiro deles foi baseado em DLL e dessa forma, o gerenciador é carregado para a memória enquanto está sendo usado. Para garantir a persistência dos dados após a DLL sair da memória, utiliza-se um banco de dados para armazenar as informações relativas à sessão. Neste método será necessário o gerenciador de sessão conectar-se com o banco de dados, todas as vezes que receber uma mensagem de um cliente. Isto poderá resultar numa degradação de desempenho.

Uma segunda abordagem de implementação do gerenciador de sessão armazena informações sobre sessão em um banco de dados, mas difere da anterior, pois uma conexão com o banco é aberta no início da sessão e é mantida durante a sua existência. Dessa forma, o número

de conexões com o banco reduz porque uma conexão será mantida durante a permanência da sessão. A terceira e última abordagem coloca o gerenciador de sessão como um processo independente, mas o armazenamento de informações sobre a sessão ocorre em tabelas na memória principal, evitando assim que ocupe o banco de dados com informações temporárias. O armazenamento em meio persistente, como banco de dados, resulta em aumento do tempo de acesso e engarrafamento na tentativa de servir a várias sessões. Por tal motivo, a opção de utilizar informações em memória pode reduzir o tempo de processamento. O problema com esta última abordagem é a confiabilidade, pois se o gerenciador de sessão cai, todas as informações em memória são perdidas, a menos que se utilize algum mecanismo de tolerância a falhas.

2.2.3 Adaptação de Conteúdo

Como consequência da heterogeneidade associada a dispositivos surge o desafio da adaptação de conteúdo. Devido à possibilidade de variações de poder de processamento, tamanho de tela, resolução de imagens, largura de banda variável e comunicação intermitente, a necessidade de adaptação de conteúdo tem presença marcante em ambientes ubíquos.

Este aspecto é também salientado como desafio em outros trabalhos que possuem a ubiquidade como foco, a exemplo de (Johnson 2005) e (Trinta 2007).

As técnicas de adaptação podem ser divididas nas seguintes categorias (Lei e Nicolas 2001):

- Adaptação à infra-estrutura técnica: Essa categoria de adaptação relaciona-se à capacidade de dispositivos e conexões de rede. Dispositivos ubíquos, em geral, variam em características técnicas, como tamanho de tela, resolução, poder computacional, etc. A adaptação deve ser capaz de prover uma apresentação apropriada a diferentes dispositivos.
- Adaptação à preferência dos usuários: Para uma apresentação multimídia específica, os usuários podem ter diferentes requisitos e níveis de detalhes de alguns parâmetros.

Quando se fala no momento da criação das mídias, as técnicas de adaptação se dividem em:

- Adaptação estática: Em tempo de elaboração, o conteúdo da informação é processado e armazenado em diferentes versões que variam em qualidade e requisitos de processamento. No momento da sua apresentação, a versão apropriada da informação é selecionada a depender do contexto do usuário.

- Adaptação Dinâmica: Neste caso, a informação é processada e entregue em tempo de execução. As alternativas disponíveis são determinadas pelo contexto específico, no momento da apresentação.

Segundo Lei e Nicolas (Lei e Nicolas 2001), a adaptação deve ocorrer em algum ponto entre a geração do conteúdo inicial e o dispositivo cliente que irá visualizar o conteúdo adaptado. Quatro tipos de localizações são possíveis: no servidor original, no *proxy*, no dispositivo cliente e no *middleware*. Cada uma das opções apresenta vantagens e desvantagens. Combinações dessas configurações podem ser úteis em algumas situações. Os principais aspectos a serem considerados para a eficiência da adaptação são a utilização de largura de banda, CPU e memória.

1. Adaptação Baseada no Servidor: O servidor é responsável por analisar o contexto e descobrir qual a largura de banda disponível. Sendo assim, a estratégia de adaptação mais adequada é selecionada. Este tipo de adaptação pode ter melhor efeito quando os criadores do conteúdo podem prever os resultados adaptados sobre diferentes condições e preferências. Por outro lado, pode complicar a implementação do servidor e dos algoritmos para geração de apresentação apropriada a uma solicitação, o que acarreta sobrecarga computacional e consumo de recursos.
2. Adaptação Baseada em *Proxy*: O cliente conecta-se através de um *proxy*, que decide e realiza a adaptação da resposta, enviando o conteúdo ao cliente. A adaptação por *proxy* reduz a carga computacional sobre o servidor.
3. Adaptação Baseada no Cliente: Depende-se principalmente da capacidade dos dispositivos. A seleção da melhor representação é realizada pelo dispositivo após receber a resposta inicial do servidor de origem. A seleção é baseada em uma lista de representações disponíveis. Essa arquitetura apresenta vantagem quando o servidor de origem está incapaz de determinar a capacidade do dispositivo, através da análise da requisição. A sua desvantagem é a necessidade de uma segunda requisição para obter a melhor alternativa de representação.
4. Adaptação baseada em *Middleware*: Permite que parte do *software* seja executada no cliente e parte nos servidores e ambos se comunicam de forma transparente. O *middleware* é responsável pela adaptação de conteúdo de acordo com o dispositivo que o usuário está utilizando.

2.2.3.1 Migração de Sessões e Adaptação de Conteúdo

De acordo com Thakolsri e colegas (Thakolsri et al. 2006), diversas abordagens suportam transferência de sessão entre dispositivos. Uma arquitetura para mobilidade de sessão e baseada no padrão IETF SIP (Rosenberg et al. 2002) é apresentada por Shacham e colegas (Shacham et al. 2005). De acordo com o protocolo *Session Initiation Protocol* (SIP), a transferência de sessão poderá se dar de forma completa ou parcial. Dependendo da necessidade do usuário, a transferência de uma sessão entre um nó móvel e um nó correspondente pode ser feita de forma completa para um único dispositivo ou particionada entre vários dispositivos. Por exemplo, um usuário pode desejar transferir uma sessão de vídeo a outro dispositivo enquanto mantém a sessão de áudio no seu próprio PDA, para que os outros não escutem a sua sessão de áudio, ou então, transferir o áudio e o vídeo para dispositivos diferentes.

A arquitetura descrita por Thakolsri e colegas (Thakolsri et al. 2006) e baseada no SIP permite 2 diferentes tipos de transferência: *Mobile Node Control* (MNC) ou *Session Handoff* (SH). No caso do MNC, o *Mobile Node* (MN), ou o nó de origem, estabelece uma sessão SIP com um dispositivo local usado para a transferência e atualização da sessão com o nó destino, *Corresponding Node* (CN), usando o SIP *session description* (Handlye e Jacobson 1998) para estabelecer sessões de mídia entre o CN e o dispositivo local. Em outras situações como por exemplo, a descarga de bateria, o usuário necessita transferir uma sessão completamente. Para isto, a arquitetura apresenta o modo SH que transfere a sessão de sinalização e de mídia para o dispositivo local.

O modelo de sessão apresentado por Thakolsri e colegas (Thakolsri et al. 2006) define um protocolo para migração de sessões nos dois modos supracitados. É utilizado o protocolo SIP combinado ao protocolo AC4 que refere-se ao processo de autenticação, autorização, contabilidade e carga. O foco da arquitetura proposta é sobretudo a contabilidade das sessões baseando-se em dispositivos de redes como servidores, roteadores e nas operadoras de redes, para viabilizar, a transferência de sessões de comunicação de um dispositivo para outro, ou para outros, dentro de um domínio, ou entre domínios de operadoras.

A relação entre adaptação de conteúdo e a migração de aplicações vem sendo explorada em outros trabalhos. A arquitetura iMASH (*Interactive Mobile Applications Support for Heterogeneous Clients*) (Phan T.; Guy e Bagrodia 2001) utiliza a idéia de migração de sessões onde é possível transferir a aplicação que está executando em um dispositivo para outro com características diferentes. Durante a transferência, estes dados podem ser filtrados para se adaptar à largura de banda disponível ou limitações do novo dispositivo.

2.2.4 Sensibilidade ao Contexto

A computação ubíqua requer sistemas e dispositivos que percebam o contexto, já que é necessário obter informações do usuário, do dispositivo e do ambiente (Schilit B.; Hilbert e Trevor 2002). Os sistemas sensíveis a contexto podem obter informações e adaptar suas operações com a mínima interferência do usuário, dependendo de uma determinada situação. Isto ocasiona um aumento na complexidade, pois é necessário que a aplicação tenha um comportamento pró-ativo, quando apropriado (Dustdar e Rosenberg 2004).

Um dos primeiros trabalhos a conceituar contexto em computação ubíqua foi realizado pelo grupo de Schilit (Schilit B.; Adams e Want 1994) que define contexto como a localização, identificação dos grupos de pessoas envolvidas e objetos próximos e as mudanças que acontecem nessas entidades. Para eles, três aspectos de contexto são considerados essenciais: onde se está, com quem se está e quais recursos estão próximos e disponíveis. Além da localização, devem ser consideradas questões como conectividade, custos de comunicação, pessoas próximas ao usuário, dentre outros aspectos.

Uma definição clássica de contexto e bastante referenciada é a proposta por Abowd e colegas (Abowd et al. 1999), para os quais Contexto é qualquer informação que caracteriza a situação de uma entidade(pessoa, lugar ou objeto) considerada relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação.

Outras definições para contexto sob o ponto de vista da computação ubíqua são apresentadas por Goularte (Goularte 2003):

- Brown e colegas (Brown P.; Bovey e Chen 1997) sugerem que contexto pode ser usado para descrever o ambiente, pessoas ao redor do usuário, situação, estado, temperatura, entre outros.
- Ryan e colegas (Ryan e Pascoe 1997) referem-se a contexto como localização, ambiente, identificação e tempo.
- Salber e colegas (Salber D. A.; Dey e Abowd 1998) definem contexto como informações sobre pessoas ou dispositivos que podem ser usados para transformar o modo como um sistema fornece serviços. Exemplos de tais informações são: dados emocionais, dados históricos, dados de localização, dados de intenção e de foco de atenção do usuário.
- Pascoe (Pascoe 1998) define contexto como o subconjunto de estados físicos e conceituais de interesse para uma entidade particular.

- Couloris e colegas (Couloris G. ; Dollimore e Kindberg 2005) afirmam que contexto de uma entidade (pessoa, lugar ou coisa) é um aspecto de suas circunstâncias físicas, relevantes ao comportamento do sistema.

Essas definições são difíceis de serem aplicadas na prática, pois baseiam-se em exemplos específicos e tendem a ser restritivas (Abowd et al. 1999), (Dey 2001) e muitas vezes limitadas à área da Computação que buscam apoiar. Dessa forma, Bazire e Brézillon (Bazire e Brézillon 2005) coletaram um conjunto de aproximadamente 150 definições de contexto, vindas de diferentes domínios e chegaram a duas conclusões principais (Vieira et al. 2006):

- o contexto atua como um conjunto de restrições que influenciam o comportamento de um sistema embutido em uma dada tarefa;
- a definição de contexto depende da área de conhecimento à qual pertence.

Embora seja difícil chegar a um consenso na definição de contexto, sobre o ponto de vista desta tese, será utilizada a definição de Dey e Abowd (Abowd et al. 1999) que afirma que contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade.

De acordo com a definição acima, o projetista da aplicação é o responsável pela decisão sobre quais informações podem ser usadas como contexto. Uma informação é considerada contexto se pode ser utilizada para caracterizar a situação de um participante numa interação (Abowd et al. 1999), (Dey 2001).

Para o ambiente UHS, por exemplo, a pessoa a ser considerada é o médico associado. O lugar relaciona-se ao hospital do ambiente onde o médico se encontra e o objeto considerado relevante para a interação é o dispositivo utilizado para a interação usuário-aplicação.

2.2.4.1 Sistemas Sensíveis ao Contexto

Um sistema é sensível ao contexto se ele usa contexto para fornecer informações relevantes e/ou serviços para o usuário (Abowd et al. 1999). Além disso, um sistema sensível ao contexto deve ser capaz de associar significado aos eventos que ocorrem no mundo exterior e usar essa informação de forma efetiva (França e Ferraz 2007).

Desenvolver sistemas sensíveis ao contexto requer mais trabalho, se comparado aos sistemas tradicionais, pois são necessárias preocupações associadas a aquisição, processamento, armazenamento, manipulação e apresentação de contexto. Vieira e colegas (Vieira et al. 2007) classificaram as tarefas associadas ao contexto em 3 etapas:

- Especificação do contexto: está relacionada a definição, especificação e modelagem dos elementos a serem considerados como contexto e as situações que eles representam em um domínio específico.
- Gerenciamento do contexto: significa prover soluções para habilitar a separação de tarefas associadas ao contexto das regras de negócios da aplicação. São necessários mecanismos para adquirir e processar o contexto de acordo com regras relacionadas a adaptabilidade do sistema e disseminar o contexto aos seus consumidores.
- Adaptação do sistema: refere-se ao uso do contexto para promover mudanças no comportamento do sistema.

As etapas de Especificação do contexto e Adaptação do sistema possuem uma alta dependência com um domínio específico e a aplicação. Sendo assim, o desenvolvedor do sistema deverá definir o domínio do conhecimento relacionado e especificar as características da aplicação que irão adaptar-se de acordo com o contexto, antes mesmo do desenvolvimento da aplicação. Diferentes aplicações em diferentes domínios demandam diferentes requisitos da especificação e adaptação de contexto. Todavia, tarefas relacionadas ao gerenciamento de contexto podem abstrair especificidades do domínio, estando pois, separadas das funcionalidades das aplicações.

Entre observar um determinado contexto até aplicá-lo efetivamente, são realizadas algumas atividades intermediárias. De acordo com diagramas de arquitetura de contextos de trabalhos anteriores (Dustdar e Rosenberg 2004), (Neto 2006), pode-se observar o diagrama de blocos conforme mostrado na Figura 2.3.

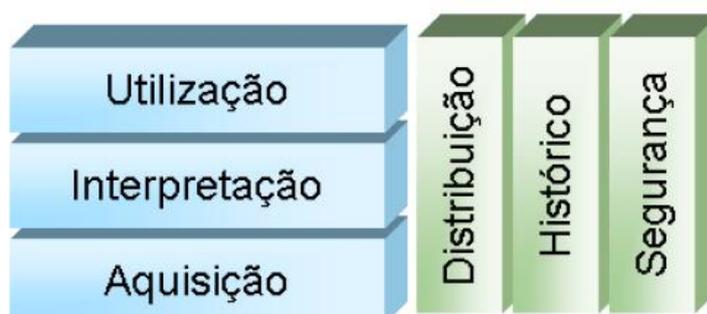


Figura 2.3 Camadas do Contexto (Neto 2006)

Na Figura 2.3 observam-se camadas verticais e horizontais. A primeira camada vertical é a de aquisição e consiste em se obter os dados de contexto. Na maioria das vezes adquirir dados

significa utilizar sensores físicos ou dispositivos que captam informações. Em geral, os dados adquiridos por essas fontes de informação precisam sofrer algum tipo de tratamento antes de serem utilizados, passando pelo processo de interpretação da informação.

A interpretação sobre dados de contexto pode ser baseada em regras e modelos probabilísticos com o intuito de elevar o nível de abstração. Uma vez que o contexto foi interpretado, ele poderá ser utilizado de diferentes formas. Para a interpretação das informações, algumas vezes é necessária a utilização de métodos matemáticos (Coutaz 2005). Devido à complexidade, a interpretação geralmente é realizada em um servidor de contexto para não sobrecarregar o dispositivo móvel do usuário, que tem recursos limitados (Xiaosheng T.; Qinghua e Ping 2006).

Em caso de utilizar informações de contexto para adaptação, a decisão por parte do desenvolvedor sobre como utilizar o contexto envolve também a decisão sobre as responsabilidades de diferentes níveis do sistema nesta adaptação (Neto 2006).

Com relação às características ortogonais da Figura 2.3, pode-se dizer que a distribuição é um aspecto inerente às estruturas de contexto, pois naturalmente são componentes distribuídos. O histórico pode ser utilizado para inferência de novos contexto. Por fim, a segurança diz respeito ao fato do contexto poder conter informações cujo acesso deva ser controlado. A segurança de um sistema sensível ao contexto representa a necessidade de proteger a privacidade do usuário.

2.2.4.2 Classificações de Informações de Contexto

A classificação adotada para o contexto aborda três aspectos: temporal, uso e tratamento da informação (Yamin 2004).

Com relação ao aspecto temporal, as informações podem ser classificadas como estáticas ou dinâmicas. As estáticas são referentes a aspectos que não se alteram com o tempo, tais como atributos relativos ao modelo/tipo do equipamento. As dinâmicas, por sua vez, traduzem aspectos do contexto e podem variar. Por exemplo, a ocupação de memória ou do processador e a localização do usuário são informações que podem variar com o tempo. Este tipo de informação pode ser obtido através de um serviço de monitoramento, que atua periodicamente ou ativado por eventos.

Com relação ao quesito uso da informação, o estado do contexto permite uso direto, quando a informação monitorada puder ser utilizada de forma bruta como informação de contexto, ou interpretada, no caso da informação puder ser processada antes de ser utilizada.

Para o aspecto tratamento da informação obtida, permite-se três classificações: corrente, no caso da informação monitorada traduzir um evento atual, podendo ser submetida a processos

como filtragem e/ou refinamento; histórica, no caso das informações monitoradas constituírem séries históricas, com o objetivo de prever o futuro; e, por fim, derivada, caso as informações de contexto possam ser constituídas pela composição de informações mais simples.

Algumas classificações de tipos de contexto também são apresentadas por alguns trabalhos anteriores tais como Goularte (Goularte 2003), Dix e colegas (Dix et al. 2000), Chalmers (Chalmers 2002) e Crowley e colegas (Crowley et al. 2002). De acordo com eles, os conjuntos de características utilizados em aplicações sensíveis ao contexto podem ser classificados nos seguintes tipos: Infra-estrutura, Sistema, Domínio e Ambiente.

Schilit (Schilit 1995) e Dix e colegas (Dix et al. 2000) representam o estado atual da infraestrutura de comunicação entre a aplicação e o dispositivo do usuário, através do contexto de infra-estrutura. Com essa representação é possível reportar, quando necessário, mudanças de estado pela ocorrência de falhas, saída ou entrada de um dispositivo no ambiente.

O contexto do sistema contém informações referentes à ciência de um dispositivo em relação aos outros nas suas proximidades e, relacionado a isto, até que ponto uma aplicação está ciente de outras aplicações que oferecem serviços (Dix et al. 2000), (Crowley et al. 2002), (Chalmers 2002). O contexto do sistema é formado pelo contexto do usuário em conjunto com o estado atual dos dispositivos e serviços utilizados. No entanto, essa visão apresenta um limite tênue entre os contextos do sistema e da infra-estrutura, já que ambos referem-se a aplicações e dispositivos.

Para Dix e colegas (Dix et al. 2000), as informações sobre a semântica do domínio da aplicação referem-se ao contexto de domínio, sobretudo considerando os relacionamentos entre os dispositivos e seus usuários e como isto pode ser usado para determinar a natureza da interface ou serviço a ser apresentado. O nível de segurança da informação e de privacidade, em situações onde informações sobre um determinado usuário (localização e atividade, por exemplo) podem se tornar disponíveis para outros usuários, também é preocupação do contexto do domínio.

O contexto do ambiente, por sua vez, refere-se a informações sobre o ambiente onde localiza-se uma entidade do sistema. Aspectos do tipo luminosidade, temperatura, localização, endereço, dentre outros são de interesse deste tipo de contexto (Goularte 2003).

2.3 Middleware com Suporte a Computação Ubíqua

Construir aplicações móveis e ubíquas sobre sistemas operacionais de rede de dispositivos portáteis é um pouco complicado e sujeito a erros. Isto torna o processo mais complexo e custoso.

Devido a tais fatores e a grande diversidade e heterogeneidade de dispositivos e de arquiteturas de redes, supõe-se que o suporte a sistemas ubíquos deva apresentar uma infra-estrutura de *software* que esconda essa não-uniformidade nas camadas inferiores. Esta infra-estrutura de *software* deve apresentar bibliotecas e um modelo de programação. É dentro desta proposta que surge o *middleware* para ambientes ubíquos, oferecendo aos projetistas das aplicações poderosas abstrações e mecanismos que os livrem dos detalhes de baixo nível.

Um *middleware* é uma camada de *software* que esconde a heterogeneidade do ambiente de execução, fornecendo um modelo de programação para facilitar o desenvolvimento de aplicações distribuídas (Tanenbaum 2001). O *middleware* é a camada que se localiza entre o sistema operacional de rede e a aplicação provendo aos desenvolvedores abstrações para esconder a complexidade do ambiente distribuído.

Em ambientes ubíquos, a sensibilidade ao contexto é um requisito importante que também deve ser levado em consideração pelo *middleware*.

A complexidade do *middleware* para fornecer informação de contexto advém das estruturas conceituais, que devem ser amplas o suficiente para tratar com diferenciados tipos de contexto, sofisticadas o bastante para fazer as distinções necessárias para as aplicações e simples para fornecer uma base prática para a programação. As informações coletadas, que caracterizam o contexto em que está ocorrendo a execução, exibem aspectos diversos e são passíveis de diferentes caracterizações (Hiltunen e Schlichting 2001).

O modelo de contexto para cada aplicação é a descrição da dependência entre o estado do contexto e sua interpretação, tanto pela aplicação, como pelo *middleware* que gerencia sua execução (Yamin 2004).

As pesquisas sobre *middleware* para sistemas sensíveis ao contexto vem sendo orientadas através do desenvolvimento de arquiteturas de *middleware* que suportam muitos paradigmas de adaptações e garantias de desempenho em se tratando de dispositivos com recursos limitados. A escolha do modelo de contexto impacta no desenvolvimento do *middleware* de três maneiras (Rocha e Endler 2005):

- O modelo de contexto pode aumentar a complexidade das infra-estruturas requeridas para a manipulação de contexto;
- O *middleware* pode usar modelagem de contexto para inferir como o contexto poderá ser usado na aplicação;
- O modelo de contexto poderá permitir a configuração dos requisitos de precisão pelas aplicações, afetando o desempenho da disseminação do contexto.

As informações de contexto precisam ser modeladas para que se tenha um entendimento de como o contexto é usado e gerenciado pelo sistema (Brézillon) (Dustdar e Rosenberg 2004) (Henricksen e Indulska 2005). Dessa forma, Strang e Linnhoff-Popien classificaram os modelos de contexto em seis tipos (Strang e Linnhoff-Popien 2004). São eles:

- Modelos de pares atributo-valor: uma informação de contexto é descrita como um conjunto de pares atributos-valor, no qual um atributo descreve uma propriedade do contexto em questão. Este é o modelo mais elementar e de mais simples implementação, mas de utilização limitada devido à ausência de tipagem e inter-relacionamento entre contextos;
- Modelos baseados em esquemas: contextos são representados como estruturas de dados hierárquicas, compostas de pares atributos-valores, e descritas por linguagens de marcadores como XML (*eXtensible Markup Language*). Diferentemente do modelo anterior, no modelo baseado em esquemas, a informação de contexto possui uma estrutura bem definida e permite a composição de informações de contexto;
- Modelos gráficos: o contexto é modelado por meio de linguagens gráficas, a exemplo de extensões de UML (*Unified Modeling Language*). Modelos baseados em UML são capazes de modelar ricas composições e inter-relacionamentos entre informações de contexto;
- Modelos baseados em lógica: o contexto é representado por meio de fatos, expressões e regras que especificam como a informação de contexto pode ser inferida ou derivada a partir de outra;
- Modelos orientados a objetos: as informações de contexto são representadas por meio de modelos de objetos, fazendo o uso de encapsulamento e reutilização, definindo o menor número de propriedades, funções e regras, simplificando a representação do conhecimento em sistemas mais complexos. O acesso a informações de contexto é provido através de interfaces;
- Modelos baseados em ontologias: contextos e seus inter-relacionamentos são descritos por meio de ontologias. Devido a sua capacidade de descrever contextos complexos, este modelo tem sido objeto de interesse crescente e foi utilizado em vários projetos, tais como CoBrA-ONT (Chen 2004) e GAIA (Roman 2002).

Strang e colegas (Strang e Linnhoff-Popien 2004) realizam um comparativo entre os modelos de contexto, tendo como base seis requisitos necessários à modelagem de contexto para

computação ubíqua: composição distribuída de informações de contexto, validação parcial da informação, riqueza e qualidade de informação, incompletude e ambigüidade, nível de formalidade e aplicabilidade. Os autores concluem que os modelos orientados a objetos e baseados em ontologias são suficientemente ricos para atender a todos estes requisitos, portanto, são os mais adequados para modelagem de contexto em aplicações ubíquas (França e Ferraz 2007).

Vários projetos adotaram a estratégia do paradigma da orientação a objetos, pelo fato de ser mais simples, de fácil desenvolvimento, e ao mesmo tempo eficiente uma vez que as linguagens orientadas a objetos são amplamente utilizadas. Outras pesquisas atuais nessa linha focam o modelo que utiliza ontologias para especificar informações de contexto e relações entre eles. O paradigma de ontologias é bastante poderoso, entretanto, é necessário o uso de um mecanismo de raciocínio que seja capaz de fazer inferências às ontologias. Este requisito poderá comprometer o desempenho no gerenciamento local do contexto em se tratando de dispositivos de recursos limitados. Neste último caso, será necessária a manutenção do mecanismo de raciocínio no servidor e não localmente, devido à escassez de recursos.

2.4 Considerações Finais

O objetivo do presente capítulo foi abordar os conceitos associados à computação ubíqua e sensibilidade ao contexto. Observa-se que a sensibilidade ao contexto é muitas vezes um requisito de ambiente ubíquos, sendo portanto, importante no cenário UHS, apresentado por este trabalho.

Por outro lado, o projeto de uma infra-estrutura de software para suportar o desenvolvimento de aplicações ubíquas é muito importante para dar continuidade e expandir aplicações inteligentes, personalizadas, adaptadas às situações de contexto e preferências ambientais, dentre outros aspectos. Sendo assim, foi apresentado os aspectos referentes a *middleware* que objetivam atender a alguns requisitos impostos por cenários ubíquos, inclusive fazendo uso da sensibilidade ao contexto.

CAPÍTULO 3

Estado da Arte

Este capítulo tem o objetivo de apontar e descrever de maneira resumida os trabalhos que abordam os temas medicina ubíqua, sensibilidade ao contexto e infra-estrutura e serviços de *middleware*.

Os três temas citados acima podem ser visualizados no esquema gráfico da Figura 3.1. Nota-se que existem quatro áreas salientadas e representadas pelos números 1, 2, 3 e 4. A área representada por 1 relaciona-se a trabalhos sobre *middleware* para medicina ubíqua. A área 2 aborda os temas *middleware* e sensibilidade ao contexto, a área 3 relaciona-se a medicina ubíqua e sensibilidade ao contexto e por fim, a área 4 trata da interseção entre os 3 temas motivadores deste trabalho: *middleware*, sensibilidade ao contexto e medicina ubíqua.

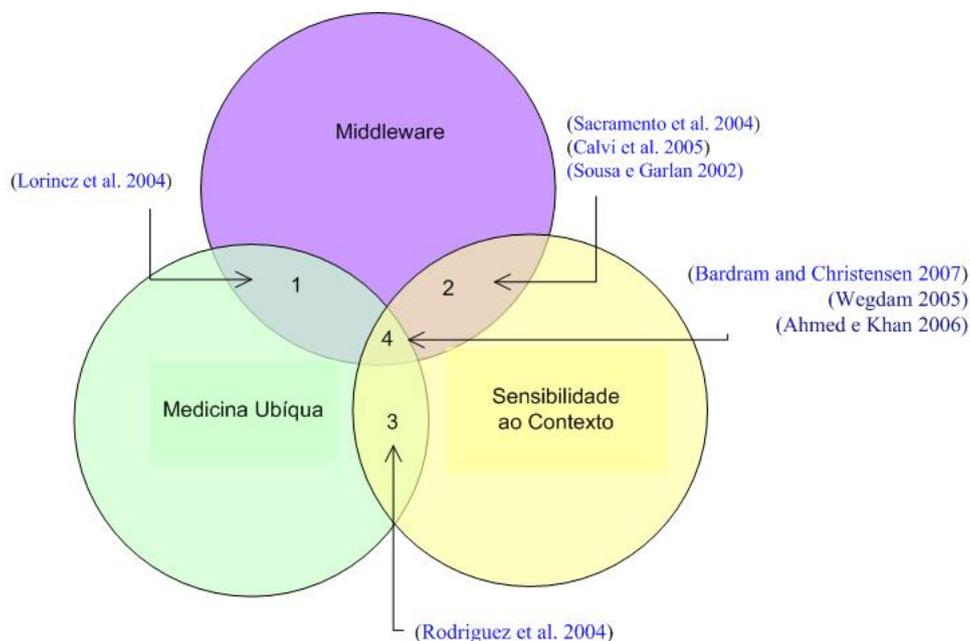


Figura 3.1 Áreas Temáticas

As seções do presente capítulo referem-se a descrições de trabalhos classificados nas áreas 1, 2, 3 e 4, respectivamente, na Figura 3.1.

3.1 *Middleware* e Medicina Ubíqua

Atualmente observa-se que a computação ubíqua vem sendo utilizada em setores relacionados a cuidados com a saúde com o objetivo de prover formas de comunicação mais fáceis e eficientes entre os profissionais que atuam nesta área.

A medicina ubíqua deve disponibilizar serviços de saúde a qualquer hora, sem restrições de localização e disponibilidade de médicos, enfermeiros e outros profissionais de saúde. Estes profissionais necessitam de ferramentas de entrega e de acesso a informações tanto no local onde encontra-se o paciente quanto no transporte dos dados a outros locais. Esta seção apresenta o *CodeBlue* (Lorincz et al. 2004), um trabalho classificado na área 1 representada na Figura 3.1, que utiliza sensores de sinais vitais.

O *CodeBlue* é uma infra-estrutura de *software* que integra nós de sensores e outros dispositivos a uma rede *ad hoc*, sem fio, oferecendo serviços de descoberta de nomes, segurança, filtragem e agregação, handoff, dentre outros. Para cenário de testes, eles utilizam o monitoramento de sinais vitais, via rede sem fio e aplicações baseadas em PDA.

O *CodeBlue* apresenta um esquema de nomes flexível. Ainda se observa a presença de *frameworks* de roteamento do tipo *publish/subscribe* e funções de autenticação e criptografia. O *CodeBlue* requer uma rede de sensores, ou seja, dispositivos com recursos altamente limitados, sendo as abordagens de *Remote Procedure Call* (RPC), agentes móveis e *Java Virtual Machine* inapropriadas para este domínio. Segundo os autores do artigo, existem vários projetos com foco em aplicações similares, para *suíte* de sensores, em vez de utilizarem um *framework* mais geral para aplicações médicas sem fio. A maioria destes sistemas são para uso em PDAs e utilizam o padrão 802.11b. As implicações de segurança são colocadas como um grande desafio, uma vez que o sucesso das redes de sensores e a sua aceitação no meio médico dependem da privacidade dos dados do paciente (Lorincz et al. 2004). A arquitetura apresentada por este trabalho pode ser vista na Figura 3.2.

O *CodeBlue* disponibiliza uma infra-estrutura de *middleware* e serviços para suportar atividades executadas em um hospital, envolvendo médicos e pacientes, dentre elas o monitoramento de paciente através de redes de sensores.

3.2 *Middleware* e Sensibilidade ao Contexto

Como mencionado previamente, a computação sensível ao contexto tem sido tratada como um paradigma promissor no que diz respeito ao atendimento das expectativas da computação

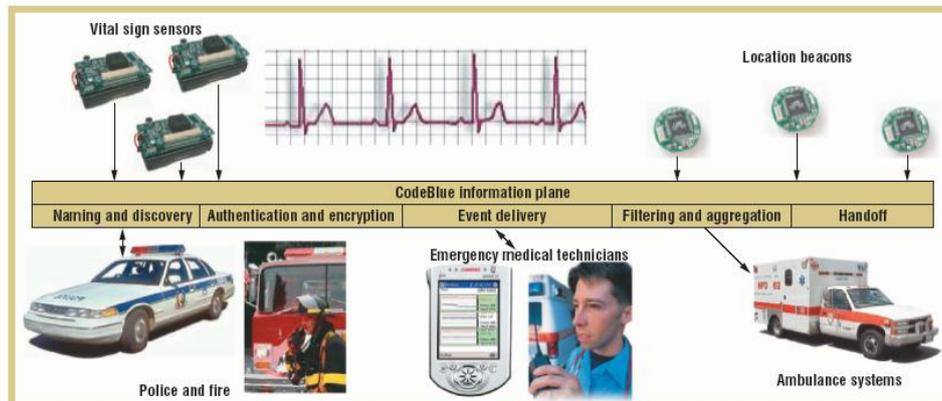


Figura 3.2 Arquitetura do CodeBlue (Lorincz et al. 2004)

ubíqua. Vários esforços têm sido desenvolvidos com o objetivo de atender aos desafios de tornar este paradigma possível, através da produção de protótipos, *middleware* e *frameworks*.

Em geral, *middleware* para sistemas sensíveis ao contexto sofrem limitações em relação à escalabilidade e dificuldade no suporte a ambientes heterogêneos (Henricksen e Indulska 2005). As atuais propostas concordam em abstrair os diferentes aspectos das soluções baseadas em contexto. Entretanto, ainda não existe um consenso sobre todos os níveis de abstração necessários para um *middleware* sensível ao contexto.

Para se manipular contexto em ambientes ubíquos é necessária a existência de uma infraestrutura de suporte de *software* adequada à manipulação de contextos diversos, provenientes de ambientes altamente dinâmicos, distribuídos, heterogêneos e em constantes mudanças. Sendo assim, pode-se salientar várias pesquisas relacionadas à construção de plataformas de serviços com suporte à execução dessas aplicações (Dockhorn 2003), (Gu et al. 2004), (Chen 2004), (Augustin 2004), (Sousa e Garlan 2002). O objetivo geral dessas plataformas é fornecer suporte arquitetural e de programação para que os projetistas possam desenvolver as suas aplicações usando serviços, mecanismos e interfaces que escondam a complexidade da manipulação de contexto.

De acordo com o esquema apresentado pela Figura 3.3, um dos requisitos fundamentais nas plataformas de suporte a contexto é a existência de um componente responsável pela tarefa de interpretação do contexto. Este componente, em geral, denominado Interpretador de Contexto, realiza a manipulação e o refinamento das informações contextuais provenientes das diferentes fontes de contexto, com a finalidade de torná-las disponíveis para uso, de forma transparente, para as aplicações (Calv C. Z.; Pessoa e Filho 2005).

O trabalho descrito em (Calv C. Z.; Pessoa e Filho 2005) apresenta um projeto de um in-

terpretador de contextos utilizado na plataforma *Infrared*. Essa plataforma dá suporte a aplicações móveis, sensíveis ao contexto. Dentre as principais características do interpretador de contexto desenvolvido por tal trabalho, destacam-se a utilização de componentes genéricos que podem ser usados (e reusados) em um grande número de aplicações sensíveis ao contexto, além de uma flexibilização quanto ao uso diversificado de linguagens de programação para descrever mecanismos de cálculos de valores contextuais e disparos de ações.

A arquitetura do *Infrared* é baseada no projeto holandês *Web Architectures for Services Platforms* (WASP) ¹, desenvolvido na *University of Twente* com o objetivo de oferecer uma infra-estrutura de apoio ao desenvolvimento e execução de aplicações sensíveis ao contexto, usando a tecnologia de *Web Services*. Também participam do projeto o *Telematica Instituut* e a *Ericsson*.

Na arquitetura do *Infrared* destacam-se os componentes com capacidade de manipulação semântica: *Service Manager*, *Registry Manager*, *User Registries*, *Task Ontologies Registry*, *Semantic Service Description Registry* e *Domain Ontologies Registry*. A arquitetura completa pode ser observada na Figura 3.3.

O interpretador de contexto utiliza uma infra-estrutura de componentes de *software* que agem como ferramentas funcionais que realizam o processo de interpretação de informações contextuais (Calv C. Z.; Pessoa e Filho 2005). Os três componentes funcionais do Interpretador de Contexto da plataforma *Infrared* são: os *Gatherers*, os *Reasoners* e os *Aggregators*. Além destes componentes funcionais, existe um componente que gerencia e organiza as atividades do Interpretador de Contexto, denominado *Manager*. Os *Gatherers* funcionam como interface entre o interpretador de contextos e o mundo externo (sensores, por exemplo), sendo responsáveis pela coleta de informações contextuais de maneira transparente. Os *Reasoners* convertem uma informação contextual mais primitiva em uma mais elaborada, tendo condições de inferir e derivar informação contextual de relevância para aplicações sensíveis ao contexto. De acordo com a necessidade da aplicação, os *Reasoners* podem variar de muito simples a extremamente complexos, podendo ser implementados por meio de diferentes estruturas computacionais. Por fim, os *Aggregators* vêm suprir a necessidade de uma unidade funcional que obtém e agrupa as informações contextuais relevantes, e que estão logicamente relacionadas para as aplicações. Através dos *Aggregators* é possível organizar as informações contextuais em uma estrutura de *software* única, fornecida à aplicação.

Embora o *Infrared* e o seu interpretador de contextos possam ser usados num ambiente de aplicações móveis sensíveis ao contexto, algumas questões ainda não foram exploradas. São elas: ambigüidades e imprecisões na manipulação de informações contextuais, propagação de

¹WASP Project: <http://www.freeband.nl/projecten/wasp/ENindex.html>

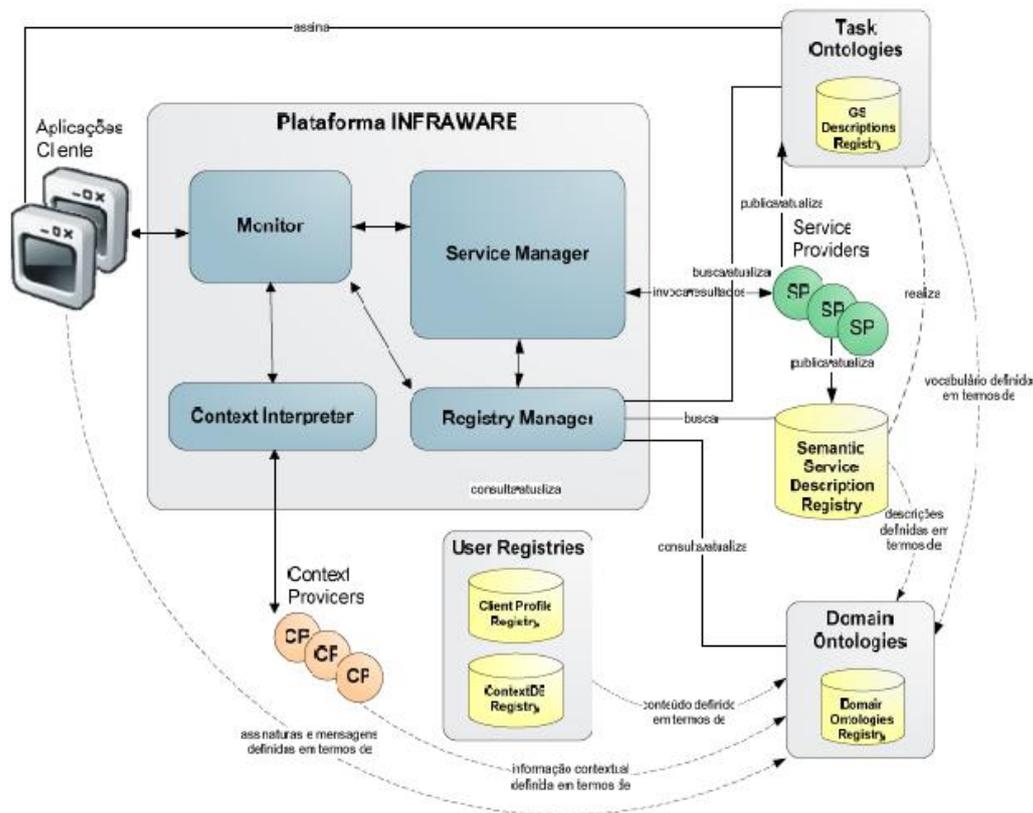


Figura 3.3 Arquitetura do Infracore (Calv C. Z.; Pessoa e Filho 2005)

incertezas acumuladas, persistência e confiabilidade das informações contextuais, entre outras.

Uma outra proposta que apresenta uma arquitetura de *middleware* para suportar aplicações sensíveis ao contexto é denominada MoCA (Sacramento et al. 2004). O *Mobile Collaboration Architecture* (MoCA) trata-se de um *middleware* para suportar o desenvolvimento de aplicações colaborativas para usuários móveis. Este *middleware* tem o objetivo de implementar uma abordagem mais flexível para gerenciamento de informações de contexto e para desenvolver abstração de contexto baseada em programação, próxima às necessidades da aplicação. A visão geral da arquitetura MoCA pode ser observada na Figura 3.4.

A arquitetura oferece alguns componentes para dar suporte ao desenvolvimento de aplicações colaborativas sensíveis ao contexto. São eles:

- O monitor é um executável que está em cada dispositivo móvel. O objetivo dele é coletar dados referentes a estados de dispositivos e de redes e enviar os dados ao *Context Information Service* (CIS). Os dados coletados envolvem carga de bateria do dispositivo móvel, uso da CPU, memória livre, ponto de acesso utilizado, lista de pontos de acessos, etc.

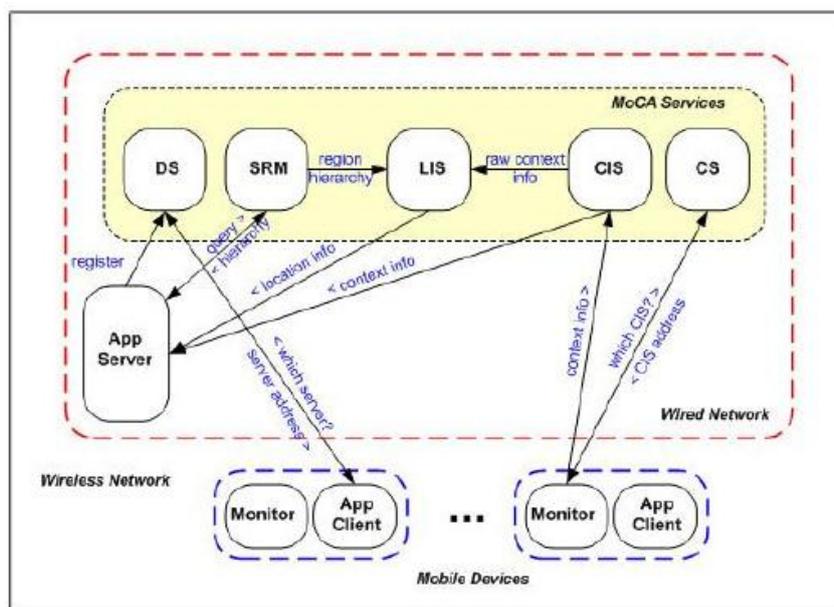


Figura 3.4 Visão Geral da Arquitetura do MoCA (Sacramento et al. 2004)

- O CIS é um componente distribuído, onde cada um de seus servidores recebe e processa as informações de contexto dos dispositivos móveis enviadas pelos monitores correspondentes. Também recebe solicitações de notificações de *proxies* da aplicação gerando e entregando eventos para eles quando ocorrer uma mudança no estado do dispositivo e esta for de interesse do *proxy*.
- O *Discovery Service* (DS) é um serviço de descoberta que armazena informações, como nome, propriedade, endereço de qualquer aplicação ou qualquer serviço registrado no *middleware* MoCA.
- O *Configuration Service* (CS) é um serviço de configuração que armazena e gerencia informações para todos os dispositivos móveis que podem usar o núcleo de serviços MoCA, como o CIS e DS.
- O *Location Inference Service* (LIS) infere a localização geográfica aproximada comparando a frequência de rádio do dispositivo, com o sinal padrão recebido pelos pontos de acessos de rede sem fio.

A seqüência de interação entre a aplicação e os serviços providos pelo *middleware* MoCA obedecem a seguinte ordem de eventos: Inicialmente o servidor da aplicação registra-se no *Discovery Service* (DS) informando seu nome e suas propriedades. Cada *proxy* da aplicação

também realiza um registro similar no DS. Dessa forma, os clientes da aplicação podem questionar o DS para descobrir como acessar um determinado serviço na rede, podendo acessá-lo através do *proxy* mais próximo. O monitor, em cada dispositivo móvel, envia o estado dos recursos locais e sinais de radio-freqüência, enviando informações de contexto ao CS. O monitor, por sua vez, obtém o endereço do *Context Information Service* (CIS) e envia informações de contexto ao CS, no momento da sua inicialização e periodicamente ao CIS.

Após descobrir um *proxy* que implementa o serviço desejado a partir do DS, o cliente pode iniciar enviando pedidos ao servidor da aplicação. Cada solicitação é roteada ao *proxy* correspondente, que processa-as, aplicando as adaptações específicas da aplicação e encaminhando ao servidor da aplicação. Por exemplo, o *proxy* poderá ser assinante no CIS com um determinado interesse, caso a memória disponível seja inferior a 10KB ou se houver troca de ponto de acesso da rede ("FreeMem < 10KB"OR "APChange = True"). Neste caso, uma notificação será gerada se uma das situações acontecerem. Quando o CIS recebe uma informação de contexto de um dispositivo através do monitor correspondente, ele checa se tem algum assinante com interesse naquele contexto. Caso positivo, o CIS gera uma notificação a todos os *proxies* que registraram interesse neste tipo de mudança.

O serviço de contexto do MoCA é composto por provedores de contexto e consumidores de contexto, tendo sido projetado para atender um conjunto de requisitos:

- Adoção de um modelo de contexto genérico e flexível que pode ser dinamicamente agregado à arquitetura do *middleware*;
- Alto desempenho, permitindo acesso rápido e disseminação de informação de contexto, evitando o uso intenso de dispositivos de rede e memória em dispositivos de recursos limitados.
- Suporte à interoperabilidade, facilitando o uso de serviços em diferentes dispositivos, sistemas operacionais e linguagens de programação.

O serviço de contexto apresenta um componente de serviço de eventos que é responsável pela entrega de informações de contexto para os consumidores interessados, através de comunicação assíncrona. O serviço de eventos adota o paradigma *publish/subscribe* e oferece uma *Application Programming Interface* (API) especializada para manipular inscrições para eventos de contexto. As variáveis de contexto utilizadas pelo MoCA são referentes a informações sobre o dispositivo, como, por exemplo, níveis de energia, memória e ocupação da CPU; a outra variável se refere a conectividade, com relação à potência do sinal dos pontos de acesso.

A modelagem de sensibilidade a contextos do MoCA suporta eventos de contexto e consultas de contexto.

O monitor do MoCA, oferece suporte para *notebooks* e *pocket PCs*. Pretende-se para o ambiente UHS, utilizar outros dispositivos móveis (*smart-phones*, por exemplo) e fixos (PCs, por exemplo). Dessa forma, o *UbiDoctor* poderá usar a API do MoCA, mas também poderá ser utilizada em estações fixas, celulares e *smart-phones*.

O trabalho descrito por Sousa e Garlan (Sousa e Garlan 2002), também referente à área 2 da Figura 3.1, apresenta o projeto AURA (Garlan D.; Steenkiste e Schmerk 2002), um *framework* arquitetural para mobilidade em ambientes ubíquos. Este projeto foi o que primeiro tratou da questão da migração de sessões. Ele define o gerenciador de tarefas PRISM, componente que coordena a migração da informação relacionada às tarefas do usuário de um ambiente para outro. O PRISM coordena os estados das tarefas interrompidas baseando-se no contexto. O AURA propõe a construção de um modelo de alto nível, independente da aplicação, para tarefas de usuários e usa este modelo para configurar e adaptar ambientes ubíquos. O PRISM implementa o controle de migração de tarefas de usuário através de uma abordagem de arquivos distribuídos com CODA ou AFS (Satyanarayanan 1990) (Satyanarayanan 1996) para gerenciamento e acesso aos arquivos ao longo do ambiente ubíquo. Para Sousa e Garlan, as tarefas consideradas na migração (Sousa e Garlan 2002) são sessões de editores de texto (MSWord) e exibição de vídeo (*Windows Media Player*).

O AURA é uma referência importante para o presente trabalho, pois foi o projeto que primeiro abordou aspectos associados à migração de sessões baseando-se no contexto. Entretanto, a abordagem utilizada pelo AURA diferencia-se da utilizada pelo atual projeto, pois utiliza a estrutura de arquivos distribuídos usando CODA ou AFS. Além disto, aspectos não funcionais associados à migração de sessão, como por exemplo, o tempo de migração também não foram analisados pelo projeto AURA.

3.3 Medicina Ubíqua e Sensibilidade ao Contexto

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

A informação requerida por um especialista é dependente da sua localização. Por exemplo, o acesso aos resultados de exames laboratoriais de pacientes deve ser mais relevante quando o

médico estiver perto do paciente que quando ele estiver em qualquer outro lugar.

Rodrigues e colegas (Rodriguez et al. 2004) descrevem um sistema de informações médicas desenvolvido para prover 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 Rodrigues trata de uma aplicação com caráter específico, mas não uma solução genérica, baseada em *middleware*.

3.4 Trabalhos Relacionados

Nesta seção, enumeram-se alguns trabalhos relacionados com a integração entre medicina ubíqua, *middleware* e sensibilidade a contexto. Tais trabalhos se posicionam na área 4 da Figura 3.1.

Os trabalhos encontrados nos levantamentos bibliográficos e relacionados ao tema são o projeto ABC (Bardram e Christensen 2007), o projeto Awareness (Wegdam 2005) e o projeto Marks (Ahmed e Khan 2006).

3.4.1 Projeto ABC

O projeto ABC vem sendo desenvolvido na Universidade de *Aarhus*, na Dinamarca e conta com a colaboração da equipe do Hospital de *Aarhus* (Bardram 2004), (Bardram e Christensen 2007), (Bardram 2003).

Os princípios do projeto ABC são resultantes de pesquisas iniciadas em 2001 e tomam como base o conceito de *Activity-Based Computing* (o que deu origem a 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 relevante para a sua tarefa (Voita 2002).

O projeto ABC vem seguindo um processo iterativo de desenvolvimento, atendendo a cinco temas que refletem algumas ações desempenhadas diariamente em grandes hospitais. São eles: controle e administração de medicamentos por enfermeiros, prescrição de medicamentos por médicos, colaboração, conferências e cirurgia (Bardram e Christensen 2007).

Bardram e Christensen (Bardram e Christensen 2001) abordam cenários do dia-a-dia dos profissionais da área de saúde, avaliando como as tecnologias de *middleware* podem prover uma forte fundamentação para soluções pervasivas e móveis. Dentre os cenários apontados pelo grupo de pesquisa destacam-se os seguintes: prescrição médica, através de discussão entre profissionais 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; procura de medicamento na farmácia do hospital; videoconferência com teleconsulta para telediagnóstico.

A arquitetura proposta pode ser visualizada na Figura 3.5. De acordo com a arquitetura, o componente *Session Service* inicia uma sessão e o *Session Manager* colabora com o *Component Manager*, na busca dos componentes corretos (*viewers e controllers*) no *Component Repository*. O *Lookup and Discovery Manager* suporta a descoberta de recursos, localização e contexto. Em cooperação com o *Awareness Monitor* em todos os clientes, o gerenciador tria as relações entre lugares, objetos e pessoas. O *Notification Manager* mantém o rastro das notificações submetidas. Este gerenciador é usado para estabelecer sessões assincronamente e colaborar com o componente de descoberta de serviços, a fim de encontrar quem está sendo notificado, onde ele está e qual é o seu dispositivo. O *Notification Service*, que é executado no cliente, é usado para submeter notificações a outros e manipular as notificações recebidas. O módulo *Security and Authentication* previne um acesso não autorizado usando listas de controle de acesso que incluem uma noção de acesso, dependente da localização e de quem está usando o dispositivo (Bardram e Christensen 2001).

Bardram (Bardram 2003) descreve a utilização de um *framework ABC* cujo objetivo é prover uma plataforma de programação para o desenvolvimento de aplicações. O projeto ABC, assim como o projeto AURA, também utiliza a idéia de migração de sessões, permitindo aos usuários transferir suas atividades de um equipamento a outro, enquanto se deslocam em um hospital. Entretanto, a transferência de aplicações entre dispositivos não contempla o uso de dispositivos móveis, mas sim a transferência para aparatos como telas e projetores em paredes para compartilhamento entre grupo de usuários.

O projeto ABC também não considera possíveis interrupções e atrasos no processo de migração de uma aplicação entre dois dispositivos, sendo estas as maiores preocupações da proposta *UbiDoctor*. Pode-se ainda ressaltar que o *UbiDoctor* aborda a mobilidade remota e a mobilidade local, conforme diferenciado por (Bardram J.; Kjær e C. 2003), enquanto o projeto ABC trata de situações envolvendo apenas a mobilidade local, onde os usuários se deslocam dentro de um hospital.

O ABC *Framework* possui os seguintes sub-componentes (Bardram 2003):

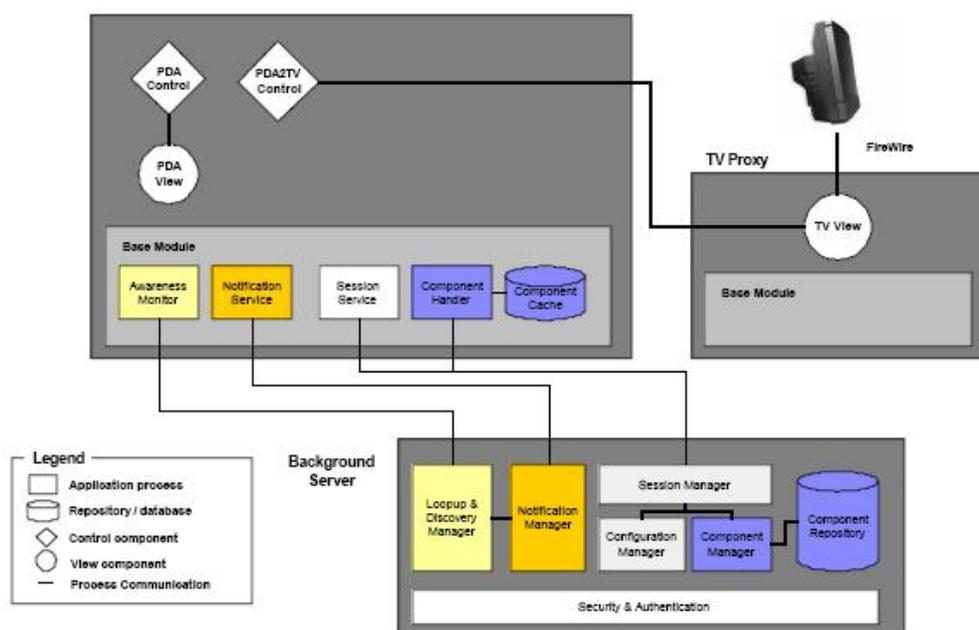


Figura 3.5 Modelo Lógico de Componentes (Bardram e Christensen 2001)

- Sub-sistema de sensibilidade ao contexto: monitora continuamente e obtém informações de contexto. Pode ser acessado de aplicações médicas ou pode ser configurado para notificar aplicações de acordo com o desejado.
- Sub-sistema de autenticação de usuário.
- Sub-sistema de colaboração: permite a participação de videoconferências.
- Sub-sistema de ciência social: este subsistema utiliza informações sobre a atividade e o contexto dos usuários.

Em um dos artigos publicados pelo grupo é apresentada uma aplicação que utiliza a sensibilidade ao contexto computacional para trabalhos médicos em ambiente hospitalar (Bardram 2004). É apresentado o projeto de um *container* de pílulas sensíveis ao contexto e uma cama hospitalar também sensível ao contexto e ambos reagem e se adaptam de acordo com alguns eventos que acontecem.

Os sistemas PEP tradicionais não oferecem nenhuma noção de sensibilidade ao contexto. A idéia de fazer uso do contexto é adquirir e utilizar informação sobre um dispositivo para prover serviços apropriados a uma configuração em particular. O objetivo do trabalho descrito em (Bardram 2004), no que diz respeito a sensibilidade ao contexto, é prover um *framework*

técnico e conceitual que pode ajudar os programadores de aplicações a criar sistemas médicos que implementem funcionalidades de sensibilidade ao contexto.

3.4.2 Projeto Awareness

Outro trabalho relacionado é o projeto Awareness (Wegdam 2005). Este projeto propõe um *middleware* geral de computação ubíqua, mas utiliza a medicina como cenários de experimentos. Através desta proposta, pacientes podem ser monitorados e tratados a distância por médicos.

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 projeto Awareness foca na infra-estrutura para a sensibilidade ao contexto que habilita a responsividade das aplicações e valida isto através de protótipos de aplicações móveis na área de saúde. Através do uso dessa infra-estrutura de *software*, torna-se possível o monitoramento de pacientes a distância que estão em situação de saúde crítica.

A arquitetura do Awareness é composta por 3 camadas: infra-estrutura de rede, infra-estrutura de serviços e aplicações móveis na área de saúde. Essa arquitetura pode ser observada na Figura 3.6.

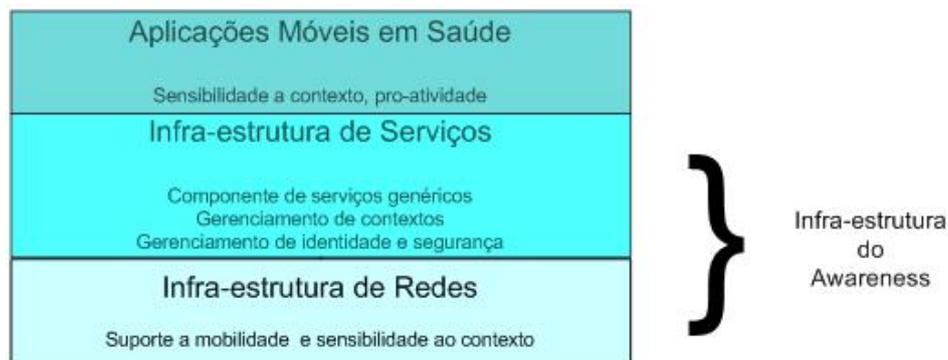


Figura 3.6 Camadas do Awareness (Wegdam 2005)

A primeira camada é responsável pelo acesso e uso das redes de comunicação, incluindo suporte à sensibilidade ao contexto em mobilidade. A camada de infra-estrutura 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 *Body Area Network* (BAN) (IEEE 2008), 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.

A proposta *UbiDoctor* segue a mesma proposta de *middleware* do projeto Awareness, porém propondo um conjunto novo de serviços, com maior ênfase no suporte à manutenção da sessão dos usuários das aplicações. Ademais, o foco das aplicações do Awareness é o tratamento de pacientes, enquanto que o *Ubidictor* dá suporte ao relacionamento entre os médicos.

3.4.3 Projeto Marks

A mesma linha de *middleware* segue o projeto Marks (*Middleware Adaptability for Resource Discovery, Knowledge Usability and Self-healing*) (Sharmin e Ahmed 2006). A arquitetura do Marks é ilustrada na Figura 3.7. O núcleo de serviços do *middleware* suporta aplicações concorrentes e descoberta de serviços seguros.

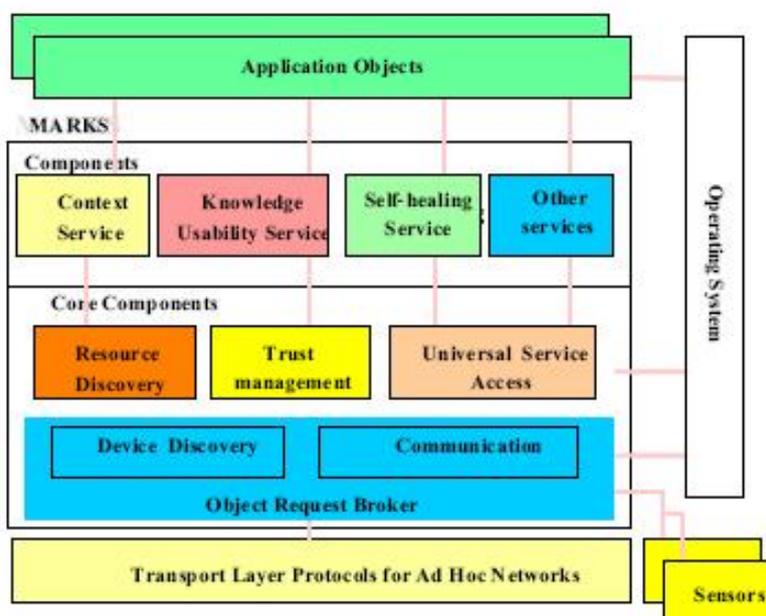


Figura 3.7 Arquitetura do Marks (Sharmin e Ahmed 2006)

Os componentes do núcleo do Marks incluem o *Object Request Broker* (ORB), unidade de descoberta de recursos, unidade de acesso a serviços universais e unidade de gerenciamento confiável. Acima destes componentes do núcleo estão os serviços oferecidos pelo Marks. São eles: serviço de contexto, serviço de conhecimento de usabilidade e serviço de auto-recuperação, além de disponibilidade para adicionar mais serviços. Os objetos da aplicação comunicam-se com os serviços e os componentes dos serviços se comunicam com o ORB. A

comunicação entre os dispositivos acontece através do ORB.

Foi desenvolvida, pelo mesmo grupo de pesquisa, uma aplicação na área médica denominada *Healthcare Aide* (Ahmed e Khan 2006), que utilizou, em sua base, os serviços do *middleware* Marks. Esta aplicação foi projetada de modo a prover mais conveniência na relação entre médicos, paciente-médico e enfermeiro-médico, oferecendo interação em tempo real entre os usuários, facilitando a tomada de decisões

Embora o Marks e o *UbiDoctor* trabalhem com aplicações em cenários de saúde, eles diferem no sentido que o Marks se propõe a tratar adaptação de conteúdo mas não a migração de aplicações, enquanto que o *UbiDoctor* propõe tratar ambos os aspectos.

Para sintetizar as diferenças entre os projetos citados e o *UbiDoctor* foi elaborada uma tabela (Tabela 3.1) contendo os objetivos dos projetos e a sua abrangência. A última coluna da Tabela 3.1 refere-se as diferenças dos projetos em relação ao *UbiDoctor*.

3.5 Considerações Finais

Este capítulo teve o objetivo de apresentar as áreas temáticas do presente trabalho de doutorado. Foram relatados trabalhos que abordam os temas medicina ubíqua e *middleware*; sensibilidade ao contexto e *middleware*; medicina ubíqua e sensibilidade ao contexto; e por fim, *middleware*, sensibilidade ao contexto e medicina ubíqua em conjunto.

Também foram apresentados os trabalhos relacionados, ou seja, que envolvem as três áreas temáticas. Os projetos envolvem o Awareness, o ABC *Framework* e o Marks, respectivamente. Observa-se que os três apresentam soluções de *middleware* e serviços para ambientes de medicina ubíqua, implementando sensibilidade a contexto, dentre outras funcionalidades. Entretanto, eles utilizam como cenário a rotina diária de um hospital, inclusive utilizando sensores, em alguns casos, e equipamentos ubíquos espalhados em alguns setores hospitalares. 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.

O Projeto ABC é, dentre todos os aqui citados, o mais relacionado com esta proposta de doutorado, entretanto o presente trabalho se diferencia do anterior nos seguintes pontos:

- Abrangência: o cenário proposto por este trabalho abrange a mobilidade remota e a mobilidade local, não se restringindo apenas à área interna de um hospital, pois pretende-se trabalhar não apenas a mobilidade de médicos no hospital mas principalmente em um contexto metropolitano.

Tabela 3.1 Síntese de Trabalhos Relacionados

Projeto	Objetivos	Abrangência	Diferenças em relação ao UbiDoctor
ABC	Prover uma plataforma de programação para desenvolver aplicações que utilizem a idéia de migração de sessões	Mobilidade local	Transferência de sessão para aparatos como telas e projetores de parede; Não se preocupa com interrupções e atrasos na migração.
Awareness	Propor um middleware para aplicações móveis na área de saúde com sensibilidade ao contexto; O conjunto de serviços propostos se destina ao monitoramento de pacientes com saúde crítica a distância.	Mobilidade local e remota	Conjunto diferente de serviços com ênfase na migração e manutenção de sessões; Trabalha com a relação médico-paciente em vez de médico-médico.
Marks + Healthcare Aide	Propor um middleware para aplicações móveis com sensibilidade ao contexto e adaptação de conteúdo	Mobilidade local e remota	Trata apenas da adaptação de conteúdo; Não se preocupa com aspectos funcionais e não funcionais da migração de aplicações.

- Atraso na migração de aplicações: visando aumentar a produtividade, existe uma necessidade do médico de que a migração de sessões ocorra com o mínimo de atraso e este aspecto ainda não foi tratado pelos trabalhos anteriores.
- Interrupção na migração: há uma preocupação de que caso haja uma interrupção no processo de migração, sem a devida conclusão, o médico possa realizar o resgate imediato da sessão, sem perda de informação, já que se trata de um sistema de informações de saúde.
- Complexidade do processo de migração: o serviço de gerenciamento de sessão é composto de vários componentes para viabilizar a atividade complexa de migração de aplicação e os problemas decorrentes da mesma citados nos dois itens acima.

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

Os serviços do *UbiDoctor*, desenvolvidos para dar suporte ao ambiente UHS, estão desacoplados de qualquer aplicação específica de medicina ubíqua, o que permite que diversas aplicações possam ser executadas dentro dessa rede de serviços e utilizem os serviços básicos disponibilizados por ela. Dessa forma a solução do *UbiDoctor* trata-se de uma solução baseada em *middleware* e serviços.

Arquitetura de Serviços *UbiDoctor*

Este capítulo apresenta a arquitetura de serviços de *middleware* para suporte ao cenário previamente ilustrado na Seção 1.4, 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 provê as seguintes tarefas:

- Suporte à adaptação de conteúdo em ambientes sensíveis a contexto, considerando diferenças nas configurações dos dispositivos e a abrangência da rede;
- Suporte ao gerenciamento de sessão, visando proporcionar a migração de aplicações¹, manutenção e persistência de sessões;
- Suporte à execução de serviços de PEP, distribuídos nos hospitais da rede, 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 4.1.

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.

¹O termo migração de aplicações (Garlan D.; Steenkiste e Schmerk 2002) envolve a passagem de uma sessão de um dispositivo a outro sem, no entanto, perder as suas características contextuais, de modo que o usuário não precise iniciar uma nova sessão no dispositivo de destino, podendo continuar do ponto em que ele estava no momento da migração.

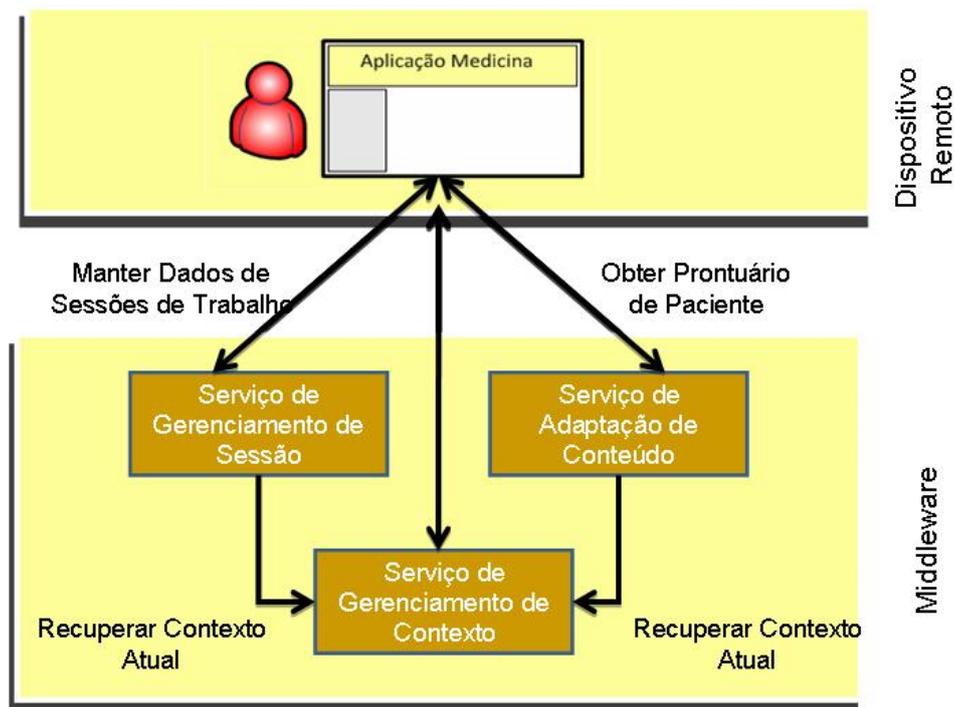


Figura 4.1 Arquitetura de Serviços do *UbiDoctor*

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. Para esta tese, optou-se por utilizar o *middleware* MoCA (Sacramento et al. 2004), já apresentado no Capítulo 3, para executar as ações desempenhadas por estas subcamadas.

As subseções a seguir apresentam os serviços representados pelo modelo.

4.1 Serviço de Gerenciamento Contexto

O serviço de gerenciamento de contexto é o componente mais abaixo na arquitetura do *UbiDoctor* e é responsável por inspecionar as informações contextuais e repassar aos demais serviços essas informações, através de variáveis de contexto. A fim de facilitar a modelagem, este trabalho adota um pequeno conjunto de informações contextuais, que são divididas em três variáveis de contexto para o cenário do UHS: dispositivo, localização e status do usuário. A idéia de contexto de dispositivo segue o conceito de contexto de infra-estrutura definido por Schilit (Schilit 1995) e Dix (Dix et al. 2000).

Com relação à variável de contexto do dispositivo, três classes são permitidas de acordo

com a Tabela 4.1. Os dispositivos selecionados foram aqueles aos quais os médicos tem mais acesso, tais como *desktops*, *notebooks*, *tablets PC*, PDAs e diversos tipos de telefones celulares, desde os mais simples aos que possuem mais recursos computacionais. Este agrupamento de classes faz-se necessário, pois a classe do dispositivo está relacionada aos seu tamanho de tela e teclado, que causa impacto na utilização do sistema por parte do usuário.

Tabela 4.1 Classes de Dispositivos

Classe	Exemplos de Dispositivos
1	Desktops e Notebooks
2	Tablets, PDAs e Smart-phones
3	Celulares e PDAs de baixo poder computacional

A variável de contexto de localização é representada por uma tupla <localização Relativa, localização Simbólica>. A localização relativa representa o código do hospital onde o usuário se encontra, caso o mesmo esteja em algum hospital da rede UHS, ou um código que representa uma área cadastrada na rede. Por exemplo, o código de localização relativa 100 pode ser referente ao "Consultório MEDCenter" ou a "Residência da Dra Luíza Barros". O código também pode ser referente a uma área indeterminada (por exemplo, 0), o que significa que o médico não se encontra em qualquer hospital credenciado, nem realizando acesso externo aos hospitais ou em local de lazer (praia, clube, shopping), usando algum dispositivo cuja localização é conhecida no UHS.

A localização simbólica diz respeito a uma localização interna a um hospital da rede, como por exemplo, bloco cirúrgico, ala da pediatria, enfermaria, refeitório, etc. Quando a localização relativa não é referente a um hospital, a localização simbólica é preenchida com o valor 0 (Zero), pois não precisará ser utilizada.

Por fim, a última variável de contexto representa o status do usuário. O médico pode estar em diferentes situações dentro do ambiente: Disponível, Ausente ou Desconectado. Os possíveis valores para a variável de status são resumidos na Tabela 4.2.

Tabela 4.2 Status do Usuário

Status	Descrição
Disponível	O médico encontra-se disponível para qualquer interação no ambiente
Ausente	O médico está distante do dispositivo utilizado para conexão
Desconectado	O médico encontra-se desconectado do ambiente

Na primeira situação, o médico encontra-se disponível para qualquer tipo de ação dentro do sistema e está conectado ao mesmo, através de um dispositivo qualquer com acesso aos

recursos disponíveis ao seu perfil de usuário. Quando o médico encontra-se com o status Ausente significa que ele estará conectado ao sistema, mas não se encontra próximo do dispositivo usado para conexão. Por exemplo, um médico está no seu consultório particular, conectado ao ambiente mas a qualquer momento ele pode ser interrompido, quando o seu próximo paciente chegar para uma consulta. A partir deste momento, ele se afastará do dispositivo para atender o paciente e seu status no sistema será ausente. No caso do médico encontrar-se desconectado, ele não poderá ter acesso aos recursos e as funcionalidades do sistema. Um exemplo de mudança de status é ilustrado na Figura 4.2.

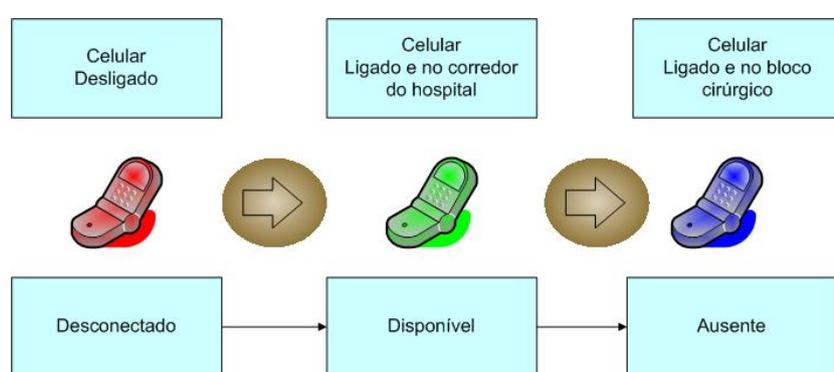


Figura 4.2 Exemplo de Mudança de Estado

Através do monitoramento das variáveis de contexto será possível passar informações aos serviços de gerenciamento de sessão e adaptação de conteúdo. As informações contextuais repassadas ao serviço de adaptação de conteúdo podem ser úteis para filtrar informações que sejam mais interessantes ao usuário baseando-se nas informações adquiridas pelo serviço de contexto. Por exemplo, o sistema não deixa o médico ter acesso a certas informações do PEP quando ele não está dentro de um hospital por questões associadas à segurança e à ética médica.

Uma situação na qual informação de contexto de dispositivo é utilizada, é observada no momento em que um médico realiza uma sessão, através do seu celular, e percebe que o mesmo não possui bateria suficiente para dar continuidade a mesma. Para isto ele solicita a transferência da sessão para o seu *notebook*. Neste momento, o serviço de contexto passa informações ao serviço de gerenciamento de sessão e este por sua vez atua realizando a migração da aplicação entre os dois dispositivos envolvidos. As informações passadas ao serviço de gerenciamento de sessão dizem respeito ao contexto atual do médico (ex. dispositivo, status do médico, localização). As informações de classe do dispositivo também serão úteis ao serviço de adaptação de conteúdo que irá realizar a adequação do conteúdo a ser mostrado na tela, para o novo dispositivo.

4.1.1 Componentes do Serviço de Gerenciamento de Contexto

Para que o serviço de gerenciamento de contexto possa realizar suas funcionalidades, foi concebido um gerenciador de contexto (*ContextManager*) que trabalha em conjunto com monitores locais, inspecionando as variáveis de contexto do usuário, a fim de passar essas informações para que o serviço de contexto possa desempenhar o seu papel. Para tal, o serviço trabalha com um monitor local que inspeciona as variáveis de contexto de dispositivo e localização. O status do médico é sugerido pelo serviço de contexto, baseado na localização do usuário (ex. médico localizado no bloco cirúrgico, status ausente), podendo ser também alterado pelo usuário da aplicação. As variáveis de contexto de dispositivo e localização são consideradas de infraestrutura (Schilit 1995). A Figura 4.3 mostra o gerenciador de contexto e os seus monitores locais no ambiente UHS.

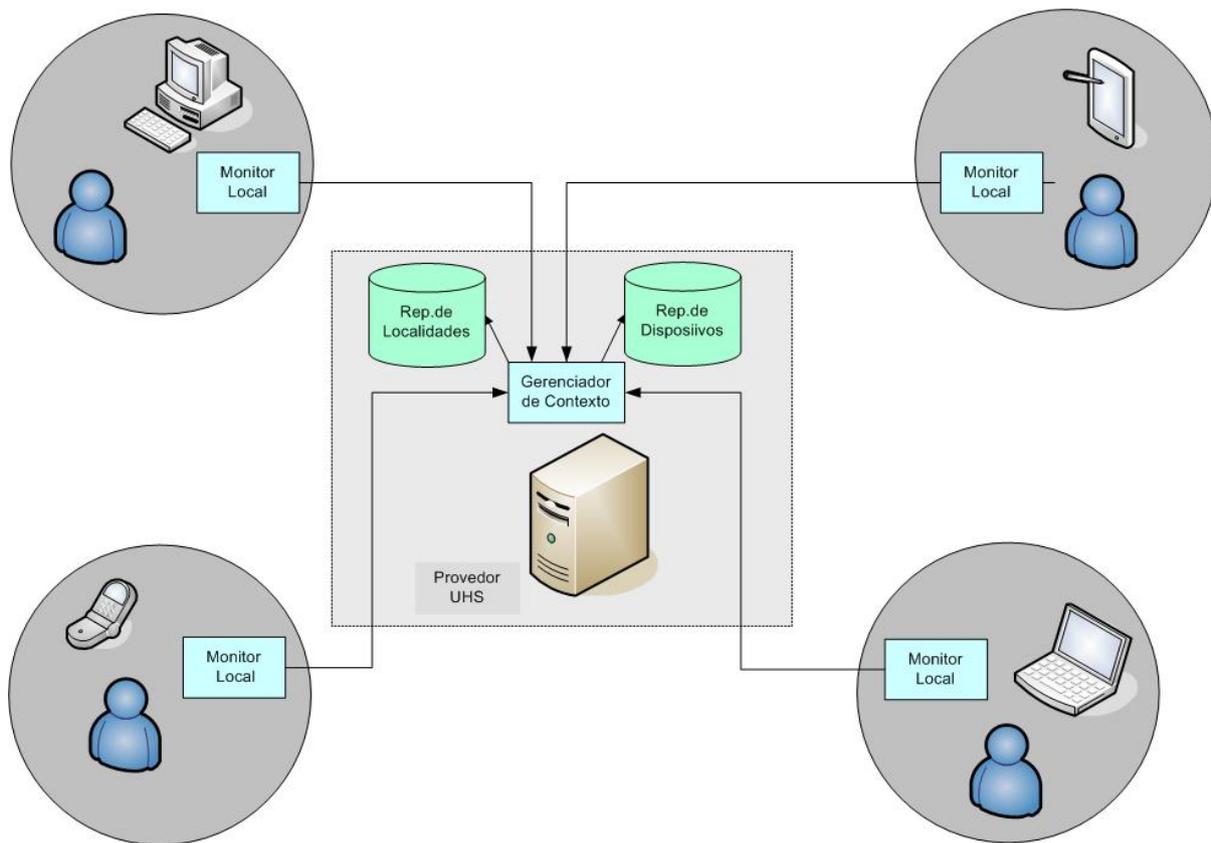


Figura 4.3 Esquema gráfico do Serviço de Contexto

O monitor local inspeciona a classe, o código identificador do dispositivo no qual o usuário está logado e o nível de bateria do mesmo. Também é papel do monitor indicar a localização do médico no ambiente UHS. Caso ele não esteja dentro de algum hospital da rede, nem em

algum ponto de localização conhecida, a sua localização recebe um código indicando que a mesma será tida como indeterminada. O diagrama de classes com os componentes do serviço de gerenciamento de contexto é apresentado na Figura 4.4.

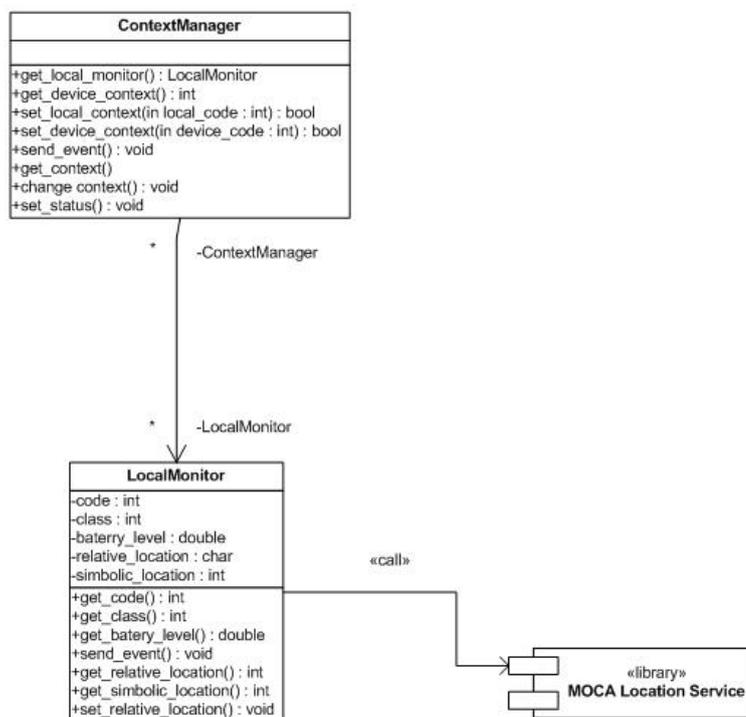


Figura 4.4 Componentes do Serviço de Contexto

O serviço de gerenciamento de contexto pode ainda dar suporte à realização da transferência de sessão entre os dispositivos de um mesmo usuário, como por exemplo, no caso de um usuário estar fazendo uma tarefa em seu *notebook* e a transfere para o seu celular, no momento em que vai deixar a sua residência. Ele poderá continuar conectado à mesma sessão do sistema, dando continuidade a alguma tarefa em seu dispositivo portátil. Os detalhes da migração serão discutidos em seções posteriores.

As mudanças de contexto também são informadas ao serviço de adaptação de conteúdo (Seção 4.3) pelo gerenciador de contexto. O gerenciador de contexto é notificado todas as vezes que algum dos seus monitores identificar mudança nas classes de contexto. Sendo assim, os monitores trabalham em relação ao gerente seguindo o modelo *publish/subscribe*.

O gerenciador de contexto é assinante de eventos referentes ao assunto "mudança de localização relativa". Por exemplo, o monitor local gera um evento quando o valor da variável de localização relativa do usuário muda de hospital A para indeterminado.

Um evento é gerado pelo monitor local no caso de baixa carga de bateria sugerindo a migração da aplicação, ou seja, que o usuário transfira sua sessão de um dispositivo portátil para outro dispositivo. Para o serviço de gerenciamento de contexto do ambiente UHS, propõe-se utilizar o serviço de notificação de eventos da infra-estrutura do MoCA (PUCRJ 2005). Essa escolha se deu devido ao fato do MoCA operar nas subcamadas de distribuição e infra-estrutura do *middleware*. Embora o serviço de notificação de eventos localize-se na subcamada de serviços comuns, ele também é disponibilizado pelo MoCA, podendo portanto, ser reusado pelo *UbiDoctor*.

A *Event-Based Communication Interface* (ECI) é a interface baseada em eventos do MoCA e provê facilidades para a implementação de comunicação assíncrona usando a abordagem *publish/subscribe*. Através dessa API, uma aplicação cliente, usando uma instância do *ECIClient*, que faz parte da API de comunicação do MoCA, pode inscrever-se como interessada em receber eventos relacionados a um assunto e usar opcionalmente uma expressão para filtrar ocorrências de eventos. O assinante (*Subscriber*) pode também publicar dados sobre um determinado assunto quando necessário. O *ECIServer*, por sua vez, é o disparador de eventos para todos os assinantes que se inscreveram como interessados nos assuntos e expressões, podendo também ser utilizado para publicar eventos diretamente do servidor de aplicação através de uma instância do *ECIServer*. De acordo com a infra-estrutura de eventos do MoCA, o assinante pode usar expressões para filtrar eventos de seu interesse. Apenas eventos com propriedades que correspondam às expressões avaliadas como verdadeiras² são entregues pelo *ECIServer*.

4.1.2 Contexto de Localização

Nos diversos trabalhos realizados sobre sensibilidade ao contexto são encontrados diversos elementos contextuais, entretanto tem merecido destaque o contexto de localização de usuários e dispositivos. Existem algumas técnicas e tecnologias para a determinação da posição física de pessoas/objetos (Trinta 2007). Essas abordagens são variáveis a depender de alguns aspectos, tais como:

- Área de aplicação: algumas técnicas podem ser aplicadas apenas em espaços fechados, enquanto outras apenas em espaços abertos;
- Precisão: podem acontecer erros de estimativas da posição de uma entidade que podem ser da ordem de centímetros até quilômetros;

²A expressão é uma *String* cuja sintaxe é baseada num subconjunto da sintaxe condicional SQL92 (PUCRJ 2005).

- Tipo de dado de localização: os dados de localização podem variar desde coordenadas globais como latitude, longitude até localizações relativas, como salas em edifícios;
- Privacidade do usuário: refere-se à permissão para fornecer informações sobre a entidade (usuário/dispositivo) a ser localizada;
- Origem da informação de localização: a informação de localização pode ser dada pelo usuário ou dispositivo e também pode ser calculada ou rastreada pelo sistema.

Conforme citado anteriormente, o serviço de gerenciamento de contexto 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. 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.

O esquema de localização do ambiente UHS utiliza repositórios de endereços MAC dos *Access Point* (AP)s dos hospitais da rede. Este repositório é denominado Repositório de Localidades. O gerenciador de contexto consulta o repositório de localidades para determinar em qual hospital o usuário está, através de uma comparação entre os endereços MAC dos APs no repositório e do endereço MAC do AP ao qual o dispositivo do usuário está conectado. Este repositório possui todos os endereços MAC dos APs do sistema e a localização relativa deles, por exemplo, "Hospital das Clínicas" ou "Centro Médico XYZ". Os hospitais já apresentam seus próprios APs e já estão cadastrados no repositório, a priori. Entretanto, nada impede que um médico cadastre os APs do seu consultório ou residência no repositório de localidades para que essa informação possa ser útil ao sistema. Este repositório é consultado pelo gerenciador de contexto para determinar a localização relativa do usuário.

Para o usuário que não está dentro de algum dos hospitais do ambiente, ou seja, não está em um dispositivo fixo em algum dos hospitais e não está conectado a um AP de hospital, e também não está usando um dispositivo fora de hospitais, mas previamente cadastrado no repositório de dispositivos, atribui-se uma localização indeterminada.

Para essa Tese serão consideradas situações onde as conexões acontecem através de redes locais, LANs e WLANs (IEEE802.11) e Internet. Para se obter o contexto de localização através do uso de uma rede GPRS é necessário utilizar uma operadora de celular que disponibilize o serviço de localização por célula ou ainda utilizar-se de um esquema de localização baseado em GPS.

Para determinar o AP ao qual o dispositivo do médico está conectado, no caso de utilizar uma rede WLAN, é utilizada a infra-estrutura do MoCA através do seu monitor e de seu serviço de contexto denominado *Context Information Service (CIS)*. Caso o usuário esteja dentro de algum hospital é possível se identificar a localização simbólica dele através do uso do *Location Inference Service (LIS)*. O LIS é um serviço que infere a posição lógica aproximada de um dispositivo. Isto acontece através de comparações de sinais de radio-freqüência(RF) recebidos em relação a sinais medidos previamente em pontos de referências dentro de um prédio, e neste caso, do hospital, seguindo a proposta do RADAR (Bahl e Padmanabhan 2000). Portanto, antes de qualquer atividade, o banco de dados do LIS deve ser preenchido com medições RF de pontos de referências desejados. O número de pontos de referências irá determinar a confiabilidade da inferência. Através do LIS, a posição do dispositivo móvel é dada por coordenadas geográficas ou por nome simbólico. Uma região simbólica representa uma área física bem definida, como por exemplo, andar, corredor, ala ou bloco.

Uma possível situação para a mudança da variável de contexto de localização ocorre quando ela muda de um valor que se refere a uma área indeterminada no ambiente (trânsito, residência, local de lazer, consultório externo a hospitais) para um valor de localização interna a algum hospital da rede. Esta situação acontece quando o médico está chegando a um hospital credenciado ao ambiente UHS. Outra situação é aquela onde o médico encontra-se saindo de um hospital credenciado. Neste caso, ele passará de uma área determinada para uma área indeterminada e terá alterado o valor da sua variável de contexto de localização. O diagrama de seqüência que apresenta esta operação é mostrado na Figura 4.5. Os casos onde o usuário se desloca dentro de um hospital são tratados pelo serviço de localização da arquitetura do MoCA.

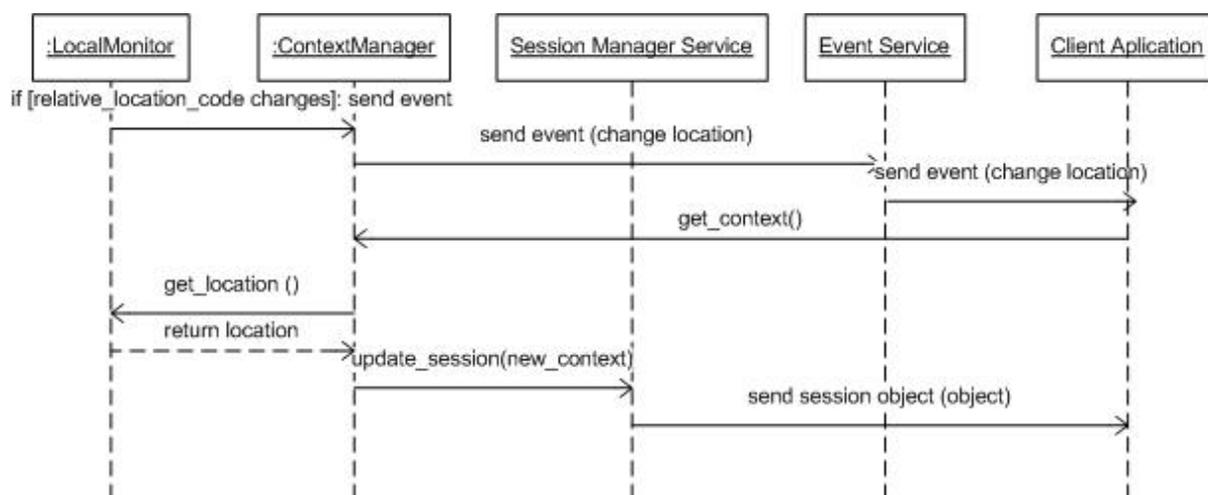


Figura 4.5 Mudança do Contexto de Localização - Saída de Hospital

Para a saída do médico do hospital, conforme observado na Figura 4.5, inicialmente o monitor local (*LocalMonitor*) reconhece que o usuário não mais está conectado àquele AP e a nenhum dos APs do hospital. Dessa forma, ele deixa de estar naquele hospital e entra numa área considerada indeterminada no sistema, mesmo continuando conectado ao ambiente. O *LocalMonitor* gera um evento que é recebido pelo gerenciador de contexto informando a desconexão e este, por sua vez, irá alterar a localização do usuário no UHS.

Dessa forma, ele percebe que há uma mudança de uma área reconhecida para uma área indeterminada. Este fato é comunicado à aplicação através do serviço de eventos (*Event Service*) e o médico é comunicado através da aplicação cliente (*Client Application*) que houve uma mudança na sua localização, no ambiente UHS. O objeto da sessão com o contexto do usuário atualizado é, em seguida, enviado à aplicação cliente.

4.1.3 Contexto de Status

Em se tratando de um ambiente ubíquo, em algumas situações o sistema pode sugerir o status do usuário baseado na sua localização, sem que o usuário precise configurar o status desejado.

Caso o usuário se encontre numa localização externa aos hospitais do ambiente UHS, o valor padrão da variável de contexto de status é ausente, pois ele pode estar em trânsito ou fazendo atendimento presencial no consultório. Dessa forma, o sistema tenta supor que o médico não pode responder ao sistema de forma imediata. Para esta primeira situação, apenas a variável de localização relativa é verificada.

Caso o usuário se encontre em determinada ala do hospital, por exemplo, bloco cirúrgico, o padrão de status é ausente, pois sugere que o médico está no sistema mas não pode responder imediatamente. Para esta segunda situação, a variável de localização simbólica é verificada para conhecer em que parte do hospital o usuário está. Embora exista essa sugestão de status baseado na localização, o usuário pode manualmente mudar o seu status para o desejado.

4.1.4 Contexto de Dispositivo

O monitor local do dispositivo identifica a variável de contexto que se refere ao dispositivo do médico. De acordo com a Tabela 4.1, percebe-se que existem 3 classes de dispositivos.

A mudança do valor da variável pode acontecer de duas formas: *hardware-driven* e *user-driven*. Na primeira abordagem, um evento é gerado pelo monitor do dispositivo informando ao gerenciador de contexto que existe uma situação de baixa carga de bateria e aconselhando que o usuário troque de dispositivo, caso o mesmo deseje continuar a sessão. O diagrama de

seqüência que retrata essa situação é apresentado na Figura 4.6. Nesta situação, o médico é avisado pelo monitor local do dispositivo (*LocalMonitor*), através da notificação de um evento ao gerenciador de contexto (*ContextManager*), que o seu dispositivo está com nível baixo de bateria. O serviço de eventos (*EventService*) utilizado para isto faz parte da infra-estrutura do MoCA. O gerenciador de contexto passa a informação ao serviço de gerenciamento de sessão (*Session Management Service*) e o médico é avisado, sendo possível transferir a sessão para algum outro dispositivo. Inicialmente, o sistema sugere alguns dispositivos cadastrados no repositório de dispositivos que possuem a localização simbólica igual a do médico. Caso não haja, serão oferecidos outros dispositivos que estejam na mesma localização relativa (no mesmo hospital) e caso o usuário não esteja em uma localização determinada, será realizada uma consulta ao histórico de sessões que contém armazenados os 5 últimos dispositivos que o médico usou para realizar uma conexão ao ambiente. Neste caso, uma lista de dispositivos associados àquele usuário é disponibilizada para que ele escolha qual deles deseja usar. O histórico de sessões é um componente do serviço de gerenciamento de sessões e será explicado na próxima seção do presente capítulo.

Em seguida, o usuário deverá ligar o novo dispositivo (caso o mesmo esteja desligado) para efetuar o login no ambiente, podendo ter sido escolhido algum dos dispositivos sugeridos ou outro escolhido pelo usuário. Essa entrada será validada pelo serviço de autenticação (*Authentication Service*) que repassará as informações do código do dispositivo e do status do usuário ao serviço de gerenciamento de sessão.

Para os casos onde o médico se encontra dentro de um hospital credenciado, o sistema sugere opções de estações de trabalho (PC, *tablet*, *notebook*) segundo a regra citada acima. O usuário escolhe a estação e a sessão fica suspensa no serviço de gerenciamento de sessão, sendo resgatada em seguida no dispositivo alvo escolhido. O serviço de gerenciamento de sessão se encarregará de atualizar o contexto do usuário na sessão, enviando o objeto atualizado à aplicação.

A segunda abordagem, parte do princípio que o usuário sinaliza que deseja trocar a sessão de dispositivo, por achar que o dispositivo de destino é mais adequado para aquela tarefa que o de origem. Neste caso, a ação parte da aplicação (*Client Application*) que irá solicitar a migração da sessão para um outro dispositivo mais adequado àquela tarefa. O serviço de gerenciamento de sessão armazena os dados da sessão em execução aguardando após o novo login do usuário. No próximo login do usuário, o mesmo irá solicitar autenticação no ambiente e o serviço de autenticação irá verificar se há alguma sessão em migração para ele no serviço de gerenciamento de sessão. Uma vez que tenha sido encontrada, o novo contexto do usuário é solicitado ao serviço de gerenciamento de contexto (*Context Manager*) que por sua

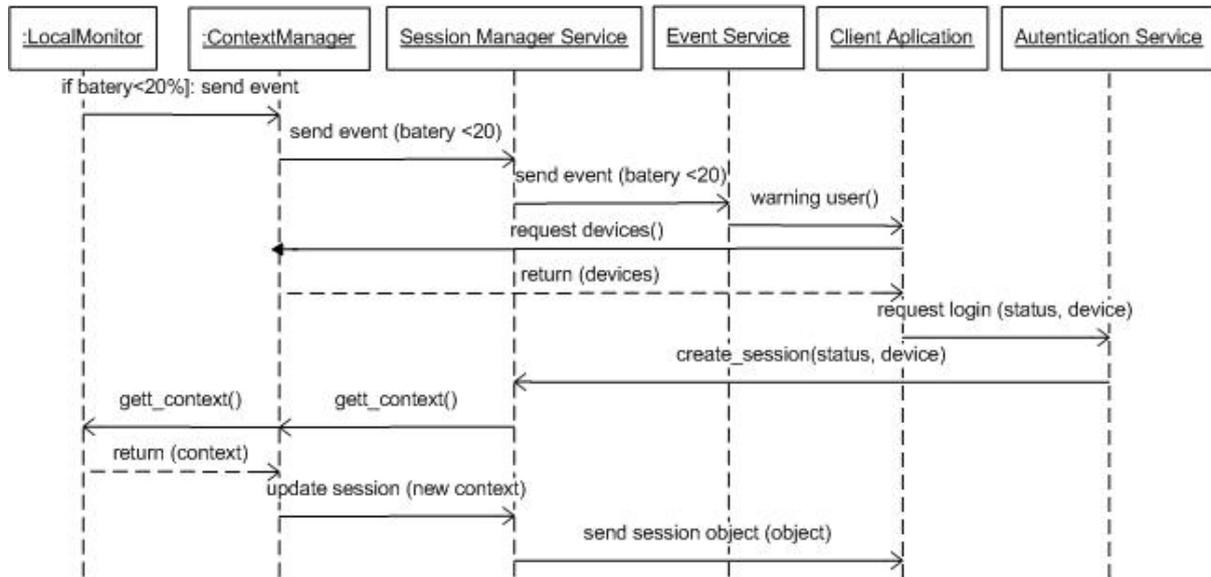


Figura 4.6 Diagrama de Mudança de Contexto de Dispositivo Hardware-driven

vez, repassa a solicitação ao monitor local (*Local Monitor*) do dispositivo. Com estes dados o serviço de gerenciamento de contexto solicita a atualização do contexto do usuário no serviço de gerenciamento de sessão e devolve para o usuário a aplicação cliente na funcionalidade ou página onde foi solicitada a migração. O diagrama de seqüência que apresenta essas atividades é apresentado na Figura 4.7.

Através do uso das variáveis de contexto apresentadas nessa seção, pretende-se minimizar problemas associados à diversidade de tamanhos de telas e configurações de dispositivos. Para isto, utiliza-se da sensibilidade ao contexto, através da interação entre os monitores e o gerente de contexto, para prover a inspeção das variáveis dando suporte às demais atividades realizadas pelos outros serviços do modelo, inclusive ajudando na solução dos problemas associados à interrupção e atrasos no processo de migração se sessões.

4.2 Serviço de Gerenciamento de Sessão

O serviço de gerenciamento de sessão do ambiente UHS é composto por seis componentes (Figura 4.8). O primeiro deles é o gerenciador de sessão, responsável pela criação e manutenção da sessão, realizando, inclusive, toda troca de mensagens do serviço de sessão com os demais componentes da arquitetura. O gerenciador de sessão se localiza no provedor do ambiente UHS, entretanto para cada dispositivo com acesso ao ambiente existe um módulo gerenciador de sessão local (o segundo componente da arquitetura) que se comunica com o gerenciador do

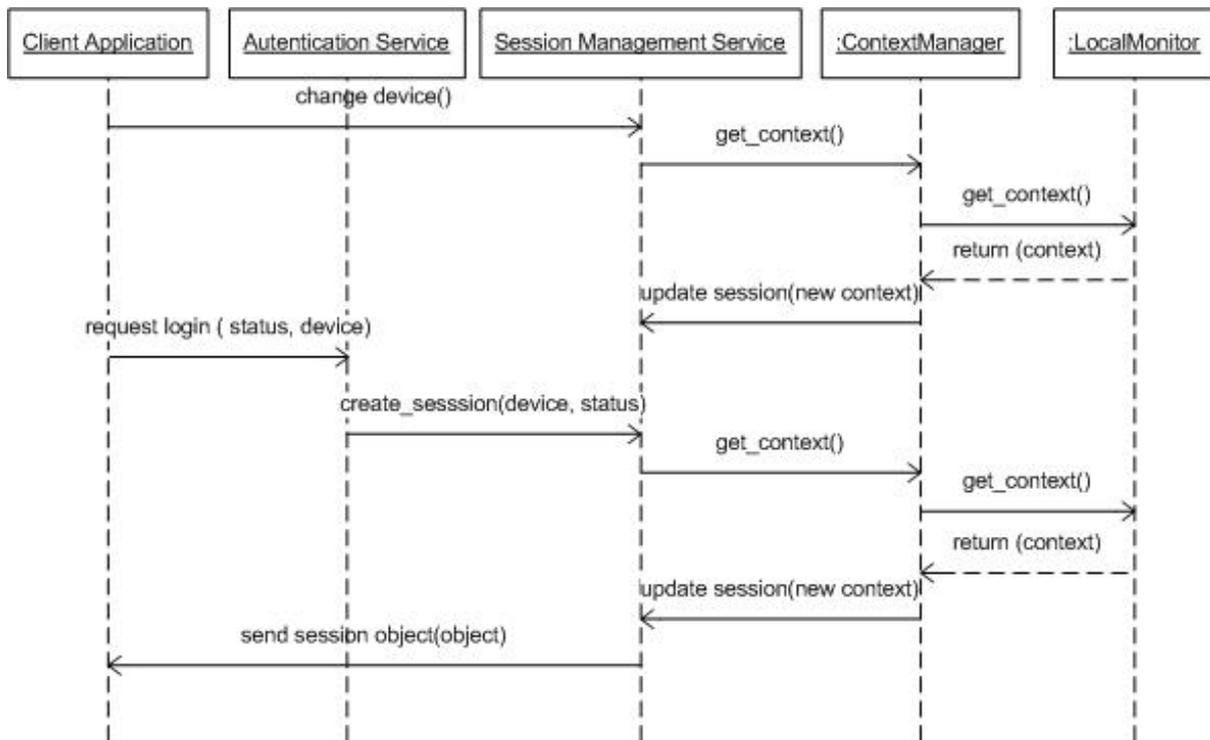


Figura 4.7 Diagrama Mudança de Contexto de Dispositivo User-driven

serviço localizado no provedor. Associada a este módulo de gerenciamento local existe uma cache local que é considerado o terceiro componente da arquitetura. Ou seja, cada dispositivo integrante do ambiente possui uma cache local.

O quarto componente é um repositório em memória no provedor, que será aqui denominado de repositório cache. Este repositório é uma área de armazenamento temporário para que as sessões sejam atualizadas, garantindo que caso haja problemas no equipamento do médico, durante a realização de uma sessão, o mesmo poderá resgatá-la total ou parcialmente através deste repositório cache. Ele também garante uma otimização de desempenho, uma vez que os dados de uma sessão podem ser armazenados temporariamente antes de serem gravados no quinto componente, o banco de dados, aqui denominado de repositório de sessões. Este último é um espaço de armazenamento que contém todas as informações referentes às sessões em execução no ambiente de rede. O sexto componente é o histórico de sessões que é um repositório que armazena as informações sobre os dispositivos utilizados pelos usuários nas últimas 5 sessões que eles participaram.

Lee e colegas (Lee et al. 2002) estudam os aspectos associados à manutenção de informações associadas às sessões, conforme citado no Capítulo 2. Existem diversas maneiras de um servidor manter certas informações requeridas no período de duração de uma sessão. Uma das

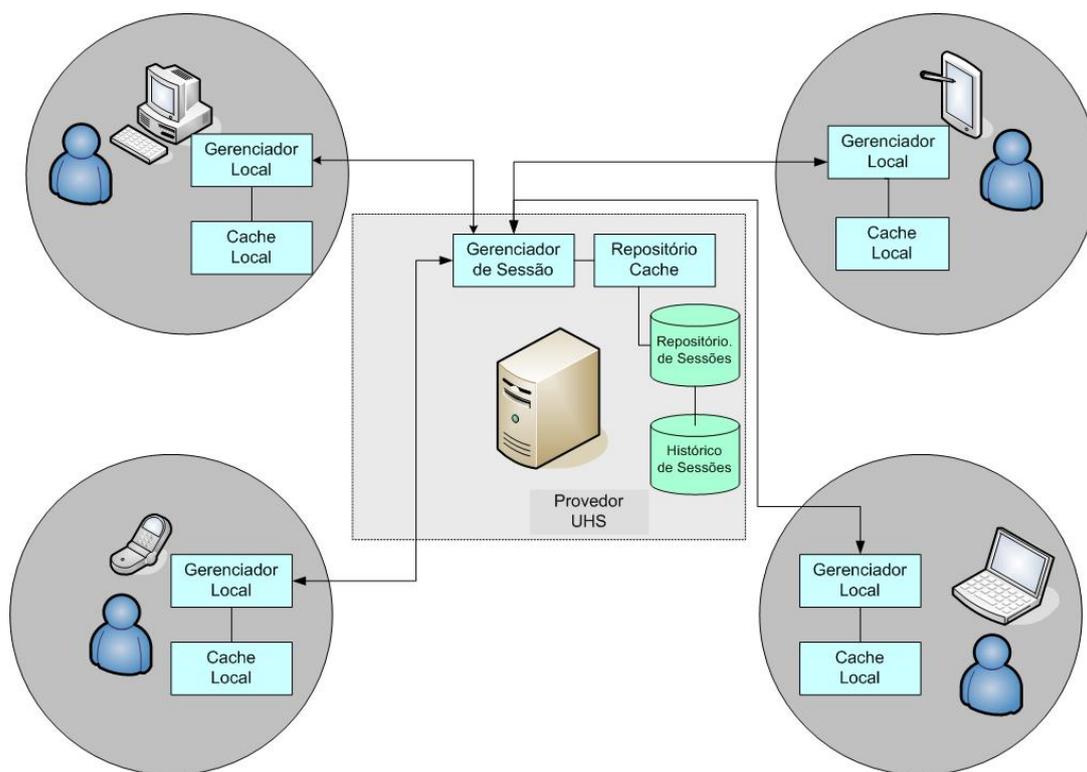


Figura 4.8 Componentes do Serviço de Gerenciamento de Sessão

formas é associando a sessão a um processo. Essa técnica pode levar à queda de desempenho devido à quantidade de processos no ambiente com alcance amplo.

É baseando-se nas alternativas propostas por Lee e colegas (Lee et al. 2002), que o gerenciador de sessão do ambiente UHS utiliza uma solução híbrida entre a segunda e a terceira abordagens apresentadas. O esquema de armazenamento de informações associadas às sessões do *UbiDoctor* apresenta informações em memória (repositório cache) para garantir melhor desempenho e apresenta armazenamento em meio persistente (banco de dados) para garantir tolerância a falhas.

No repositório, as informações referentes às sessões são armazenadas na forma de objetos. Quando uma sessão é criada, no momento de login de um usuário no sistema, por exemplo, o gerenciador de sessão instancia um objeto para representá-la. O objeto que representa uma sessão contém todos os dados associados à mesma e um identificador que a diferencia de todas as outras sessões existentes no sistema.

O Serviço de Gerenciamento de Sessões apresenta 2 tipos de sessão: o primeiro tipo representa sessões simples, onde um usuário se conecta ao sistema ou faz solicitações aos PEPs que compõem a rede provedora. O segundo tipo de sessão, denominada composta, representa

sessões de chat (conversação) entre dois ou mais médicos com acesso a dados de PEPs e troca de informações envolvendo conversas síncronas entre multiusuários.

O diagrama de classes para as informações de sessão é ilustrado na Figura 4.9. Observa-se que o componente gerenciador de sessão é o responsável por criar e manter a sessão durante a sua existência. Ele gerencia o conjunto de sessões do ambiente UHS. A sessão, por sua vez, é composta pelos seguintes atributos: identificador único, tipo da sessão (simples ou composta), tempo de início, lista de médicos participantes e status da sessão.

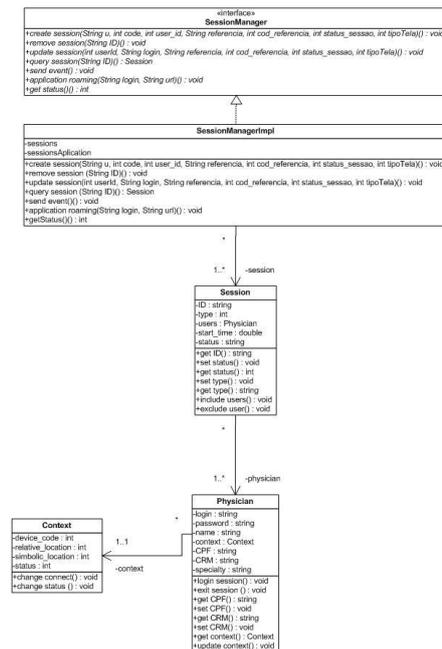


Figura 4.9 Diagrama de Classes para Informações de Sessão

Uma sessão é composta por um ou mais médicos (*Physician*) e cada um deles apresenta um contexto (*Context*) associado que, como mencionado anteriormente, envolve dispositivo, localização e status. Além dessas informações, um médico possui um nome, CPF, CRM, especialidade e dados para o seu login, como usuário e senha.

O status da sessão indica qual é a sua situação atual no ambiente UHS. O ambiente tenta manter armazenada apenas uma sessão de cada usuário. Isto significa que se um usuário está participando de uma sessão simples e ingressa numa sessão composta, a sua sessão simples será automaticamente finalizada e o mesmo passará a compor a sessão composta. A variável de status de uma sessão poderá assumir os seguintes valores: 1 (ativa), 2 (concluída), 3 (em migração) e 4 (pendente). Uma sessão é considerada ativa quando a mesma está em execução e apresenta todos os seus usuários com o status de disponível ou ausente. Para que a sessão possa ser considerada concluída, ela deverá ter sido finalizada por todos os usuários presentes,

no momento do seu término, usando os procedimentos normais de conclusão. Uma sessão ficará mantida no banco de dados com o status de concluída até que todos os seus usuários tenham se conectado numa outra sessão. Quando isto acontece, o registro daquela sessão no banco será removido. Por este motivo, há uma tentativa de manter apenas uma única sessão para cada usuário, mas não existe garantia que isto sempre aconteça.

Para evitar que sessões muito antigas sejam mantidas no banco de dados porque um usuário (que está de férias, por exemplo) nunca mais se conectou ao ambiente, o sistema utiliza uma rotina que realiza a limpeza do banco após 32 dias (30 dias de férias + margem de segurança).

Uma sessão estará em migração quando algum dos seus usuários estiver transferindo-a de um dispositivo a outro. E por fim, uma sessão pendente é aquela que pelo menos um dos seus usuários desconectou-se de forma inesperada, ou seja, sem chamar a rotina normal de desconexão.

4.2.1 Políticas de Atualizações de Dados das Sessões

Como mencionado anteriormente, existem 4 mecanismos de armazenamento de dados das sessões no ambiente UHS. A cache local, o repositório cache, o repositório em banco de dados e o histórico de sessões. Os dispositivos conectados ao ambiente UHS armazenam temporariamente as informações trocadas nas sessões na sua cache local. É importante salientar que a cache local é uma área de armazenamento em disco, pois garante a persistência dos dados mediante falhas de dispositivo (bateria e hardware em geral).

Para garantir otimização no uso da cache local, o espaço reservado para a cache em disco tem um tamanho limitado e utiliza uma política de substituição de dados para sobrescrever dados na cache quando a mesma estiver totalmente ocupada. Tradicionalmente, as informações são removidas da cache baseadas em algoritmos de expiração. Tais algoritmos determinam que informações serão utilizadas nas políticas de remoção de dados na cache, tais como tempo de uso, tamanho e histórico de acesso. Algoritmos conhecidos são o *Least Recent Used* (LRU) e *Least Frequently Used* (LFU). Algoritmos de cache mais modernos que fazem uso de dispositivos móveis são baseados em recursos temporais, localidade e prioridade (Mahgoub e Ilyas 2004).

O histórico de sessões é um repositório que possui armazenadas as informações sobre os dispositivos utilizados pelos usuários nas últimas 5 sessões que eles participaram. Estes dados são úteis, pois podem ser utilizados pelo serviço de gerenciamento de contexto para sugerir dispositivos no momento da migração.

Para o ambiente UHS, o mecanismo de atualização de dados do repositório cache com

dados da cache local utiliza valores pré-configuráveis de espaço de armazenamento local e do servidor. Em geral, a configuração do espaço para cache local é sugerido pelo gerenciador local, instalado no dispositivo do usuário. A configuração do espaço no repositório cache é definida pelo administrador do sistema através do gerenciador de sessões do servidor. Após encher este espaço, o gerenciador local envia os dados armazenados ao gerenciador de sessões do provedor e este atualiza os dados do repositório cache no servidor. Após essa atualização, o gerenciador de sessões do servidor envia uma confirmação ao gerenciador local para que o mesmo possa esvaziar a cache do dispositivo. Essa liberação de espaço em disco é necessária, tendo em vista que boa parte dos equipamentos envolvidos possuem recursos computacionais limitados. Após preencher o espaço do repositório cache, o gerenciador de sessões do servidor realiza a atualização do banco de dados do repositório de sessões.

Durante essas atualizações de dados, alguns problemas podem ocorrer. Na atualização do repositório cache com dados da cache local, poderá haver a queda do cliente antes ou após o envio de dados armazenados na cache local. Isto poderia se dar devido à intermitência da comunicação, queda de bateria ou problemas de *hardware* ou *software* no dispositivo. Para minimizar o problema da queda de bateria, o sistema já avisa o usuário desta possibilidade, mas o mesmo poderá continuar trabalhando e no entanto, ignorar o fato de que a bateria poderá acabar durante a execução da sua sessão. Caso essa queda aconteça, as seguintes situações poderão ocorrer, no momento da reconexão do médico:

- O médico retorna à sessão usando o mesmo dispositivo e a sessão ainda está em andamento;
- O médico retorna à sessão usando um outro dispositivo e a sessão ainda está em andamento;
- O médico retorna quando a sessão na qual ele participava não está mais ativa (pode usar o mesmo ou outro dispositivo);

Quando se falam em sessões compostas (interação entre médicos e acesso ao PEPs), a primeira situação acontece quando o médico trocou a bateria do seu equipamento e voltou a se logar nele. Neste momento, o serviço de gerenciamento de sessão do servidor constata que ele estava participando de uma sessão e que houve uma saída abrupta, sem ter sido solicitada pelo usuário. Sendo assim, o gerenciador local comunica-se com o gerenciador de sessão do servidor, solicitando a atualização dos dados. Este, por sua vez, solicita aos gerenciadores locais dos demais participantes da sessão da aplicação que atualizem o repositório cache do servidor e o repositório de sessões no banco de dados. Após ter sido feita essa tarefa, o gerenciador de

sessões do servidor envia ao gerenciador local do dispositivo afetado o novo objeto da sessão atualizada e o usuário volta à sessão. Uma situação similar acontece quando o dispositivo do médico é acometido por problemas de intermitência de comunicação. Após a sua re-conexão e estabilização do enlace, o mesmo procedimento para atualização dos dados acontece.

A segunda situação, ou seja, o médico retorna a uma sessão composta usando um outro dispositivo, será tratada na subseção referente à migração de aplicação. Os cenários que representam essas situações podem ser observados na Figura 4.10. O retorno a partir do mesmo dispositivo está representado no quadrado intermediário (identificado por 2) da figura e o retorno do médico ao sistema a partir de um outro dispositivo é ilustrado no quadrado mais a direita (identificado por 3) da Figura 4.10.

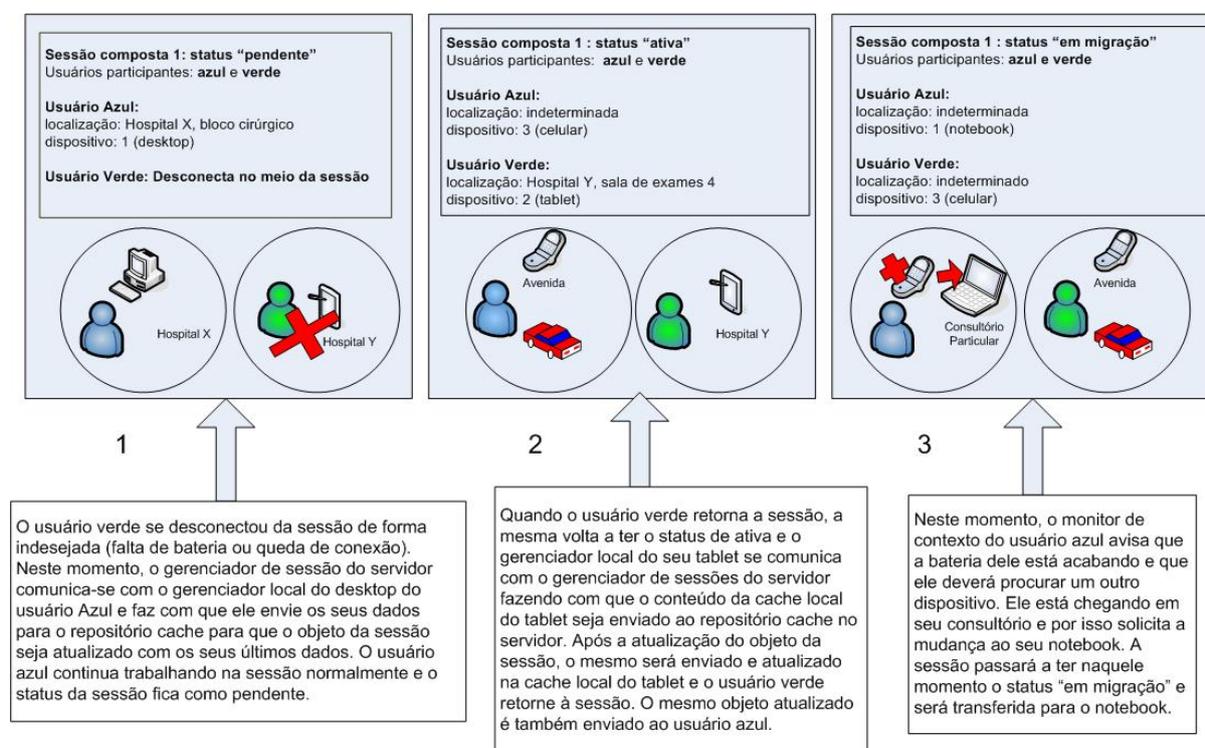


Figura 4.10 Cenário para Sessão Composta

Por fim, o médico poderá retornar após a sessão que ele participava ter sido concluída pelos seus colegas. Esta situação pode ser observada na Figura 4.11. Neste caso, no momento do login do usuário, o gerenciador de sessões do servidor verificará que não há sessão aberta na qual aquele usuário fazia parte. Entretanto, ele verificará que existiu uma sessão onde aquele usuário fazia parte, mas o status do usuário estava pendente na sessão. O usuário será questionado se deseja resgatar a sessão concluída para que ele possa ter acesso ao resultado concluído pelos seus colegas ou não, optar por iniciar uma nova sessão. Apenas os médicos presentes na

hora do fechamento da sessão serão responsáveis pelo parecer.

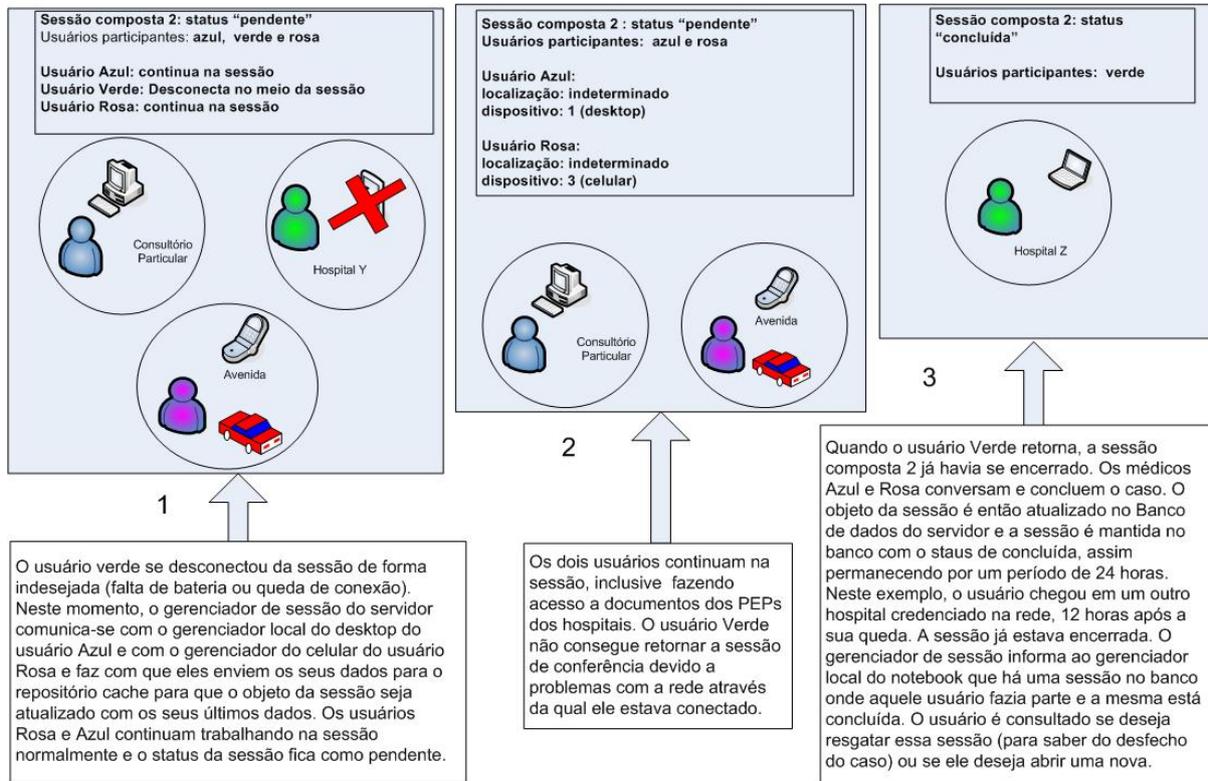


Figura 4.11 Cenário com Retorno após o Término da Sessão

Para as sessões simples, ou seja, apenas um usuário fazendo acesso ao ambiente, os mesmos problemas de interrupções podem acontecer. Os dados alterados pelo usuário são armazenados na cache local e após preencher o espaço máximo permitido, ocorre a atualização do repositório cache do servidor, ou seja, existe um espaço pré-configurado na cache local para os dados da sessão do usuário. Quando a cache local está preenchida, o gerenciador local comunica-se com o gerenciador de sessões do servidor, atualizando os dados do repositório cache e após receber confirmação que as informações foram gravadas no repositório cache sem problemas, a cache local poderá ser esvaziada para uma nova gravação de dados. Todas essas tarefas acontecem de forma transparente ao usuário.

Para os casos onde ocorre a interrupção de sessão de forma involuntária, devido a problemas no dispositivo ou na rede e o usuário retorna do mesmo dispositivo, os dados da cache local estão mais atualizados que os da cache servidora, então o usuário terá acesso a estes dados no seu retorno ao sistema e em seguida os dados locais serão atualizados na cache servidora e posteriormente no banco de dados. Isto acontece porque o gerenciador de sessões local se comunica com o gerenciador de sessões do servidor, no momento do login do usuário, consta-

tando que havia uma sessão para aquele médico, naquele dispositivo e que não foi finalizada. A Figura 4.12 ilustra esta situação. Neste caso, o usuário conecta seu celular no carregador ao chegar em um restaurante, conforme observado no quadrado de número 3.

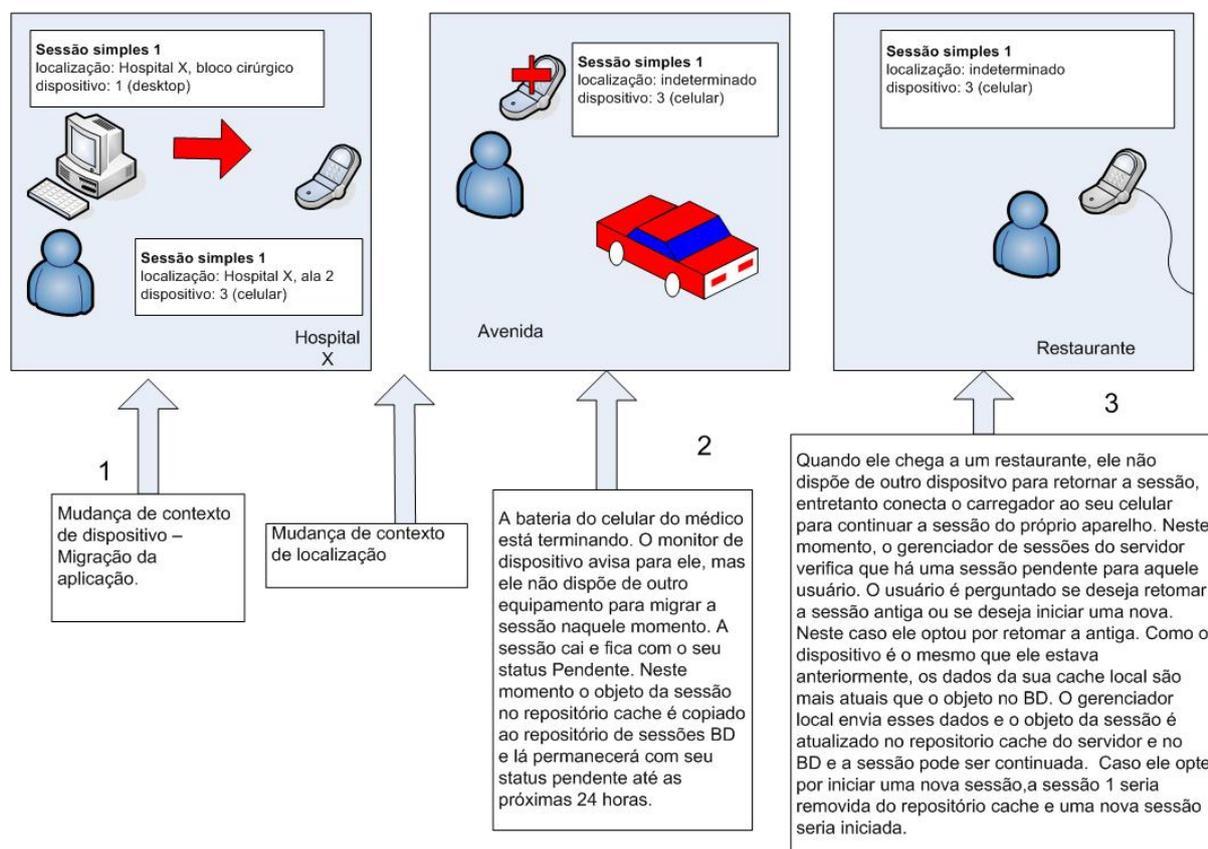


Figura 4.12 Cenário Sessão Simples

Nos casos onde o usuário médico retorna de outro dispositivo, considera-se que há a transferência de sessão (vide Subseção 4.2.3). O terceiro caso não se aplica à sessão simples, pois o usuário é o único participante da sessão. Dessa forma, esta situação recai nas duas situações anteriores.

O processo de atualização do banco de dados do repositório de sessões com os dados da cache servidora se dará após o preenchimento do repositório cache. O valor de ocupação do repositório cache até que seja necessária a sua atualização no banco de dados irá depender da escolha do administrador do sistema e será configurado através do gerenciador de sessões do servidor no momento da sua instalação.

Todas as vezes que uma sessão é atualizada no banco de dados, o objeto que a representa é enviado à aplicação cliente. Este processo acontece através dos gerenciadores de sessão, no servidor, e os locais, nos próprios dispositivos.

Acredita-se que através da implementação de duas caches, uma local e outra remota o *middleware* possa garantir uma transparência ao usuário da aplicação com relação à manutenção dos seus dados. Isto agrega valor no quesito confiabilidade do ambiente, uma vez que envolve trabalho desenvolvido por uma equipe médica e que deve se apresentar como o mais perto do trabalho real, desenvolvido de forma presencial.

4.2.2 Atividades do Serviço de Gerenciamento de Sessão

Esta subseção descreve algumas das importantes atividades desempenhadas pelo serviço de gerenciamento de sessão. As ações envolvendo o serviço de gerenciamento de sessão podem ser iniciadas pela aplicação cliente (*ClientApplication*) ou pelo serviço de gerenciamento de contexto, através do gerenciador de contexto (*ContextManager*). Observa-se tais ações no diagrama de caso de uso da Figura 4.13.

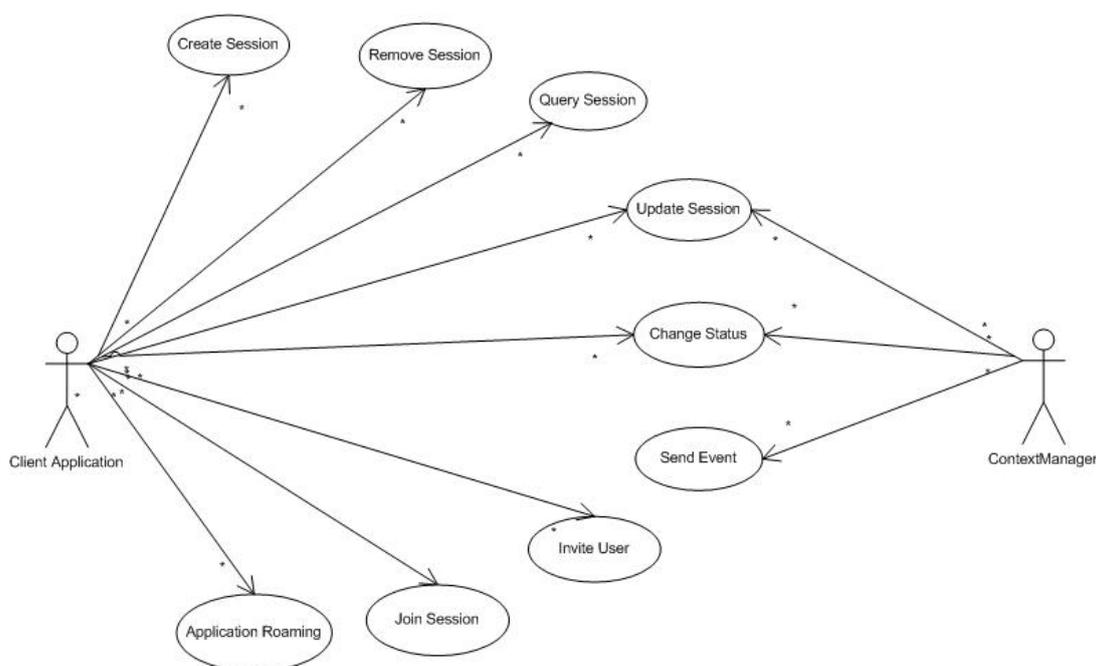


Figura 4.13 Caso de Uso do Gerenciamento de Sessão

A primeira ação representada no diagrama de caso de uso refere-se à entrada do usuário no ambiente UHS. A entrada de um usuário no sistema está relacionada à abertura de uma nova sessão ou o resgate de uma sessão pendente. Essa sessão vai possuir um identificador e será representada por um objeto no repositório de sessões. O diagrama de colaboração que traz este passo a passo é mostrado na Figura 4.14.

Quando o médico entra no sistema, através de um de seus dispositivos (PDA, celular, *no-*

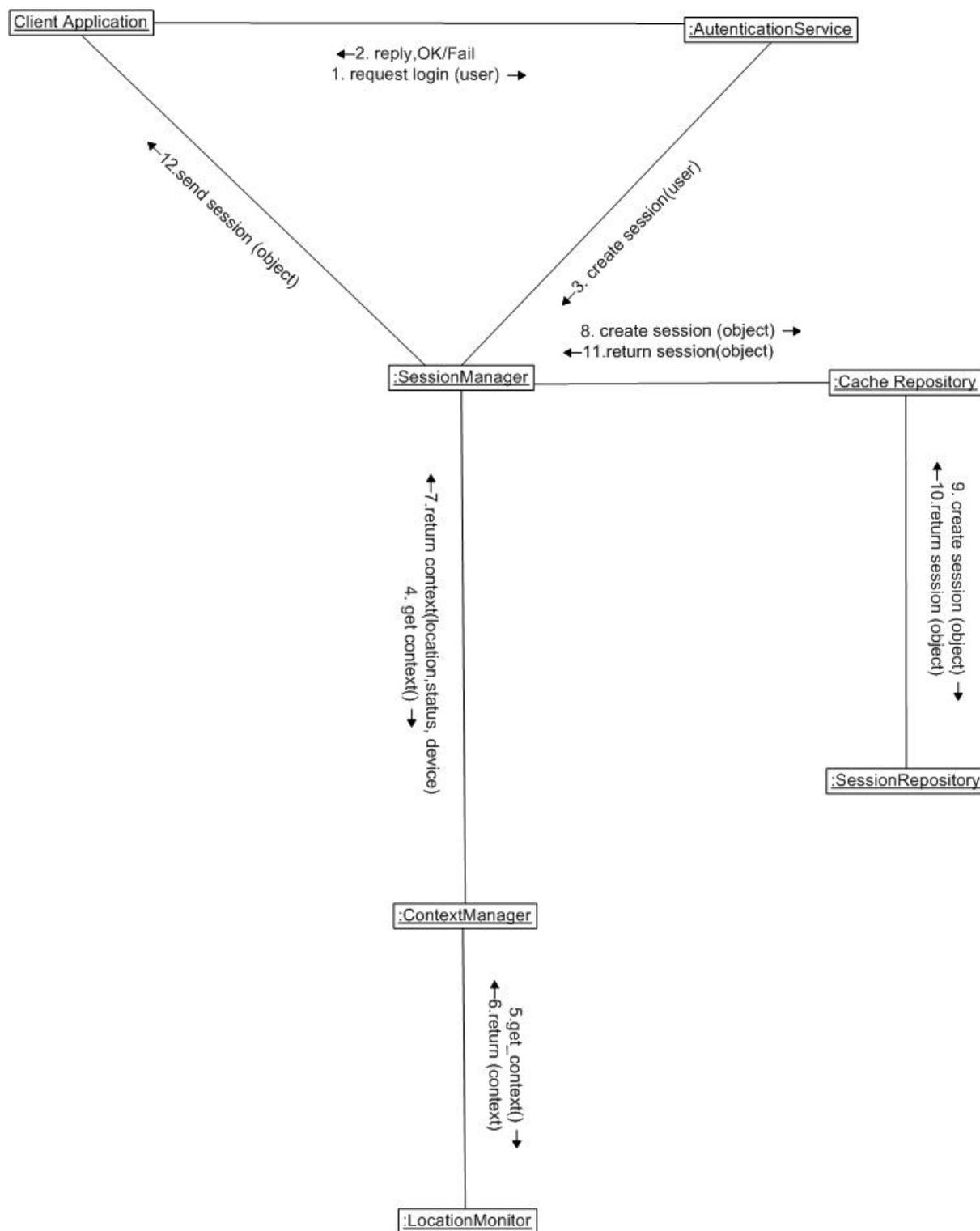


Figura 4.14 Diagrama de Colaboração - Entrada no Sistema

tebook, desktop, tablet), ele passa seu login, senha e seu status (Passo 1) que será considerado como uma das variáveis de contexto, citado em seção prévia. Estes dados são encaminhados ao serviço de autenticação de usuários que provê o mínimo de segurança no acesso a dados de modo a garantir que nenhum médico, que não seja cliente do ambiente, possa ter acesso ao mesmo.

Em seguida, todos os dados do usuário são encaminhados ao gerenciador de sessão, componente do serviço de gerenciamento de sessão responsável por concentrar as ações de criação e manutenção de uma sessão (Passo 3). Neste momento, o gerenciador de sessão deverá identificar se existe alguma sessão pendente para aquele usuário. Para isto ele executará uma rotina automática que verifica todas as sessões do servidor que possuem, no campo dos médicos participantes, aquele login de usuário. Caso positivo, uma mensagem será enviada ao serviço de autenticação e este dará a permissão para que o médico escolha se deseja retornar àquela sessão ou iniciar uma nova. Caso ele deseje retornar à sessão anterior, seu nome voltará para a lista de usuários da sessão e a mesma será resgatada. Caso ele não deseje resgatar a sessão anterior, o usuário nega o interesse e uma nova sessão é criada para ele, tendo então seu nome excluído da lista de usuários da mantido temporariamente em memória, no repositório cache, e na cache local em cada dispositivo participante da sessão.

Além deste conjunto de informações passadas pelo serviço de autenticação, um outro grupo de informações também será passado ao gerenciador de sessão. O monitor local é responsável por passar os valores das variáveis de contexto: localização, dispositivo e status do usuário, baseando-se na sua localização. Tais valores contribuem para a criação do objeto que representa aquela sessão. Por fim, o objeto criado é retornado ao usuário e o médico interagirá no ambiente, dentro da sessão recém-criada (Passo 11 e 12).

É importante ressaltar que no caso do usuário escolher retornar a uma sessão pendente, uma troca de dados precisará ser realizada entre o gerenciador de sessão local do médico e o gerenciador de sessão global do provedor.

Na verdade, as informações contidas na cache local dos dispositivos dos médicos participantes estão atualizadas e o gerenciador de sessões do servidor deverá solicitar ao gerenciador de sessões local, de cada um dos usuários envolvidos, as suas informações. O gerenciador local deverá fazer uma agregação destes dados e atualizá-los no repositório cache e no banco de dados do serviço de gerenciamento de sessões.

A atualização do conteúdo se dá seguindo a abordagem *Pull*. Isto significa que os gerenciadores locais enviam os dados atuais ao gerenciador global, respondendo ao seu pedido de atualização. Após a atualização do servidor, o objeto da sessão seguirá ao gerenciador local do médico re-entrante e este terá a sessão atualizada em seu dispositivo, podendo retornar a

partir daquele momento. Durante estes processos de atualização, os demais médicos terão suas sessões paradas momentaneamente (bloqueadas) até que o objeto da sessão atualizada possa ser também recebido pelos seus gerenciadores locais. Isto tudo acontece de forma transparente aos médicos e garante a consistência dos dados da sessão.

Outros casos de uso referentes à manutenção da sessão (*Remove Session*, *Update session*, *Query session*) também são representados no diagrama da Figura 4.13. A opção de *Update Session* pode também ser iniciada pelo gerenciador de contexto, uma vez que as mudanças de contexto podem ser inicializadas pelo monitor do dispositivo.

O diagrama de caso de uso também retrata atividades relacionadas à mudança de contexto da variável de status, solicitada, neste caso, pelo ator Aplicação Cliente (*Client Application*). Este caso de uso é representado no diagrama por *Change Status*. Essa situação acontece quando o usuário não está de acordo com o status atribuído a ele pelo serviço de gerenciamento de contexto, baseado na sua localização.

A ação de envio de eventos (representada por *Send Event*) no diagrama é iniciada pelo ator gerente de contexto (*Context Manager*) e refere-se aos fatos onde são anunciadas as mudanças de contexto e as mesmas devem ser avisadas à aplicação.

O caso de uso representado por *Join Session* (Figura 4.13) envolve a ação na qual o usuário foi convidado por alguém a participar de uma sessão e sinaliza aceitar o convite (através do clique de um botão, por exemplo). O *Invite User* já mostra a situação na qual o médico deseja convidar algum colega que esteja no sistema, em uma sessão simples, por exemplo, para participar de uma sessão composta com ele. As mudanças de dispositivos são representadas por um caso de uso específico denominado *Application Roaming* e será tratado na subseção a seguir.

4.2.3 Migração de Aplicações

Como mencionado anteriormente, este conceito envolve a passagem de uma sessão de um dispositivo a outro sem, no entanto, perder as suas características contextuais, de modo que o usuário não precise iniciar uma nova sessão no dispositivo de destino, podendo continuar do ponto em que ele estava no momento da migração.

Para o escopo do ambiente UHS será utilizado o modo SH (*session handoff*) onde a transferência se dá integralmente de um dispositivo para outro (Thakolsri et al. 2006).

No *UbiDoctor*, a migração pode se dar devido a diversos motivos. Para este trabalho, classifica-se a migração em duas categorias: *Hardware-driven* ou *User-driven*. Essas categorias serão abordadas nas subseções a seguir.

4.2.3.1 Migração *Hardware-driven*

Na categoria *Hardware-driven* estão as ações que avisam da necessidade da migração devido a alguma restrição de *hardware*, como por exemplo, a escassez da bateria. Como apresentado na seção anterior, neste caso, um aviso é originário do monitor de contexto, passando pelo gerenciador de contexto e pelo gerenciador de sessão. O diagrama de seqüência que representa a migração devido ao baixo nível de bateria já foi apresentado na Figura 4.6.

Como mencionado, existe a possibilidade da bateria do dispositivo acabar antes do processo de migração ter sido concluído ou até mesmo pode haver uma queda no enlace de comunicação com o dispositivo de origem e a migração não acontecer por completo.

Para os casos de interrupção da migração, duas situações podem ocorrer: o usuário realiza a sua reconexão de um outro dispositivo ou o usuário realiza a reconexão do mesmo dispositivo. Esta última situação já foi tratada previamente. Para a reconexão de um outro dispositivo, as seguintes situações podem acontecer:

- Para sessões simples, durante o seu login, o serviço de autenticação comunica-se com o gerenciador de sessão e este identifica que tem uma sessão pendente para aquele usuário. Ele é então questionado se deseja continuar e se sua resposta for negativa, a sessão será apagada do repositório cache do servidor e será iniciada uma nova sessão. Caso ele deseje continuar a sessão, o gerenciador de sessão do servidor irá solicitar ao gerenciador local do novo dispositivo que carregue os dados do repositório cache para a cache local e será dada a continuidade à sessão. Entretanto, nesta última situação poderá haver a perda de dados que haviam sido salvos apenas na cache local do outro dispositivo, caso a queda se dê antes do processo de atualização do repositório cache no servidor acontecer. Caso a queda aconteça após a atualização do repositório cache, os dados são resgatados do repositório cache, na íntegra. Dessa forma, caso existam perdas, ela será mínima, já que o sistema dispõe de mecanismos de atualizações automáticas.
- Para as sessões compostas que envolvem mais de um usuário e com acessos a dados de PEPs, por exemplo, o usuário que saiu da sessão de forma inesperada recebe a informação do serviço de autenticação, no momento do seu próximo login. Neste caso, o serviço de autenticação se comunicou com o serviço de gerenciamento de sessão e constatou que havia uma sessão na qual aquele usuário participava e que ele saiu de forma abrupta. Essa informação é repassada ao médico e ele é questionado se deseja continuar a fazer parte daquela sessão. Caso ele diga não, ele será excluído da lista de usuários da sessão e uma nova sessão será aberta para ele. Caso ele responda positivamente, o gerenciador de sessão do servidor irá solicitar aos gerenciadores locais as informações das caches locais

e atualizará o repositório cache e o repositório em banco de dados, no provedor, dando continuidade à sessão. Nos casos de sessão composta, caso o usuário retorne quando a sessão já tenha sido finalizada pelos demais participantes, a mesma ficará armazenada no repositório de sessões até no máximo 24 horas após a queda.

A Figura 4.15 ilustra a situação onde, numa sessão simples, o usuário retorna à sessão através de um outro dispositivo. Neste caso, o médico poderá também conectar seu celular num carregador, ao chegar em seu consultório particular, e então os dados que estavam na cache local do celular poderão ser enviados ao *desktop*, desde que usuário concorde, no momento em que ele é questionado sobre essa situação. Isto é um artifício que evitaria que ele perdesse as informações presentes na cache local do celular e que ainda não haviam sido atualizadas no repositório de sessões do servidor, antes do usuário desconectar-se involuntariamente do sistema. É fundamental que a migração do Quadro 1 da Figura 4.15 aconteça dentro de 51 segundos (Tentori e Favela 2008) para que o médico tenha a migração de sua sessão efetuada enquanto desloca-se de uma atividade para outra, de modo a possibilitar melhorias de produtividade em sua rotina de trabalho.

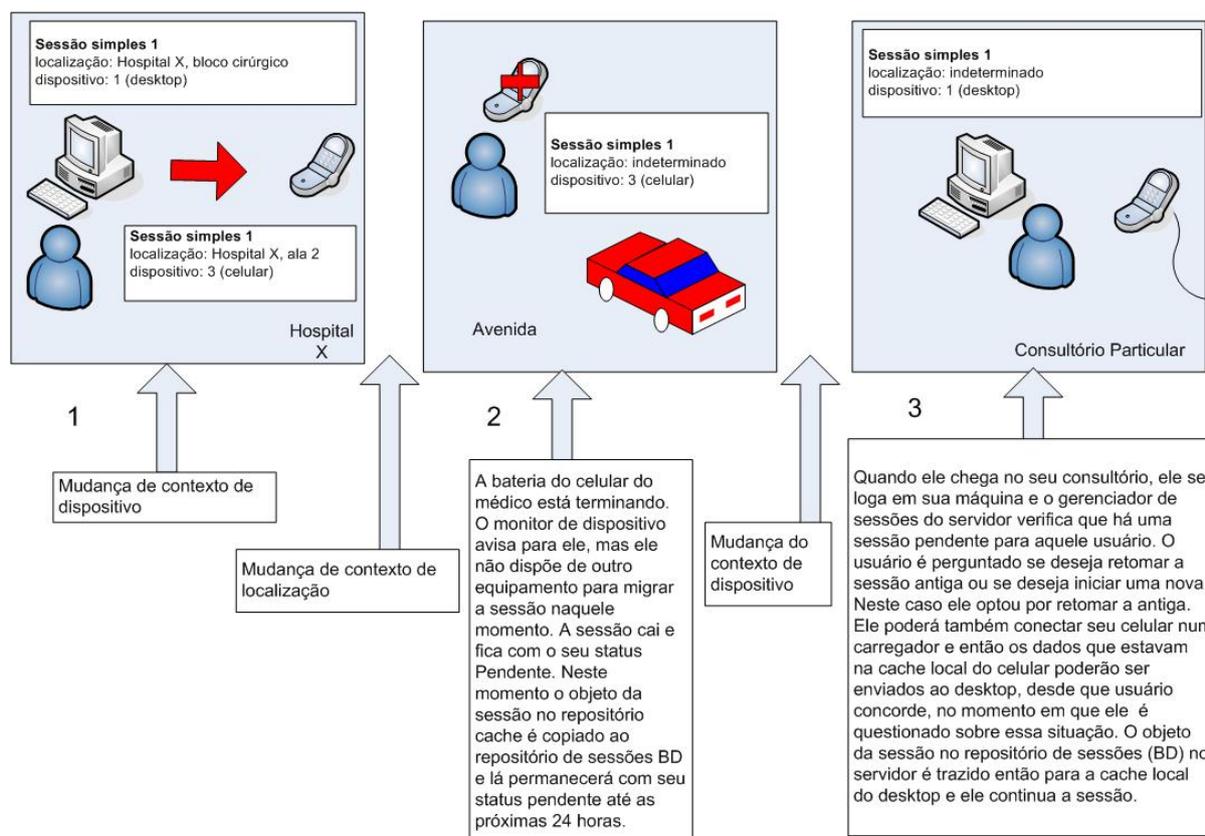


Figura 4.15 Cenário Sessão Simples com Migração

4.2.3.2 Migração *User-driven*

Na categoria *User-driven* estão as ações que são iniciadas pelo próprio médico, onde ele sinaliza que deseja mudar o seu dispositivo naquela sessão. Isto poderia acontecer, por exemplo, quando um médico é convidado a participar de uma sessão e o mesmo encontra-se em seu consultório, conectado ao sistema através do seu celular. No decorrer da sessão, ele precisa acessar exames de Raio-X ou ultrassom e prefere dar continuidade a mesma usando o seu *desktop*. Neste caso, a manifestação inicial parte do usuário da aplicação e não do serviço de contexto. Entretanto, ainda assim observa-se o uso da computação ubíqua, salientando-se que apesar de ser uma ação solicitada pelo usuário, continua existindo a possibilidade do uso de computação a qualquer hora, em qualquer lugar e usando diversos dispositivos.

O diagrama de seqüência que retrata essa situação é apresentado na Figura 4.7. O tempo de migração também deverá ser mínimo para possibilitar ganhos de produtividade ao médico, enquanto realiza a troca de atividades.

Problemas com interrupções de migração também podem acontecer e as mesmas soluções são adotadas para minimizar as perdas. A perda poderá ser zero, caso a interrupção da sessão ocorra após a atualização dos dados no repositório cache e poderá ser mínima no caso da cache local não ter sido salva no repositório cache.

4.3 Serviço de Adaptação de Conteúdo

O termo adaptação de conteúdo relaciona-se à modificação da representação de objetos ou conteúdos, visando atender às capacidades de manipulação de mídia dos dispositivos e às restrições impostas pelos enlaces de comunicação (Buchholz e Schill 2003). Essas modificações podem incluir transcodificação de formatos (XML para WML, JPEG para WBMP), redimensionamento de imagens, conversão de mídia, omissão ou substituição de partes de documento ou fragmentação de documentos. Conversões semânticas, tais como, traduções de linguagens são também consideradas operações de adaptação de conteúdo.

O desenvolvimento de variados dispositivos ubíquos, como PDAs, telefones celulares, *palm-top*, *tablet* e *laptop*, consolidou a necessidade de adaptar os conteúdos que trafegam nas redes. Estes dispositivos geralmente possuem características próprias como tamanho de tela, memória, processamento e aplicativos. A diversidade de tipos de acesso também é um fator motivador para a adaptação de conteúdo, pois as redes podem variar de grande largura de banda (ex. conexão a cabo, *Asymmetric Digital Subscriber Line* (ADSL) e conexão a rádio), a outras com largura de banda variável (ex. *Wireless Local Area Network* (WLAN)). Dessa forma,

percebe-se que um dos problemas a serem tratados por um serviço de adaptação de conteúdo é a heterogeneidade, uma vez que alguns dispositivos contam com limitações de *hardware* e tamanhos de telas e muitas vezes o conteúdo a ser apresentado por eles pode ser prejudicado.

Com a possibilidade de migrar sessões entre os dispositivos utilizando informações de contexto, provenientes de um serviço de gerenciamento de contexto, a ação de adaptação de conteúdo torna-se muito importante.

Um aspecto essencial na realização de tarefas associadas à adaptação de conteúdo é a definição de uma política de adaptação, que decide qual a melhor operação a ser realizada sobre o conteúdo, quando e para quem solicitá-la. Caso seja necessário realizar mais de uma adaptação, essa política define ainda a ordem de execução das adaptações. Entretanto, para desempenhar essa política, devem ser consideradas informações sobre o dispositivo do usuário e o conteúdo, ou seja, informações de contexto. Dessa forma, o serviço de adaptação de conteúdo também utiliza informações gerenciadas pelo serviço de gerenciamento de contexto para a realização de suas ações. O diagrama de seqüência representando a adaptação de conteúdo pode ser observado na Figura 4.16.

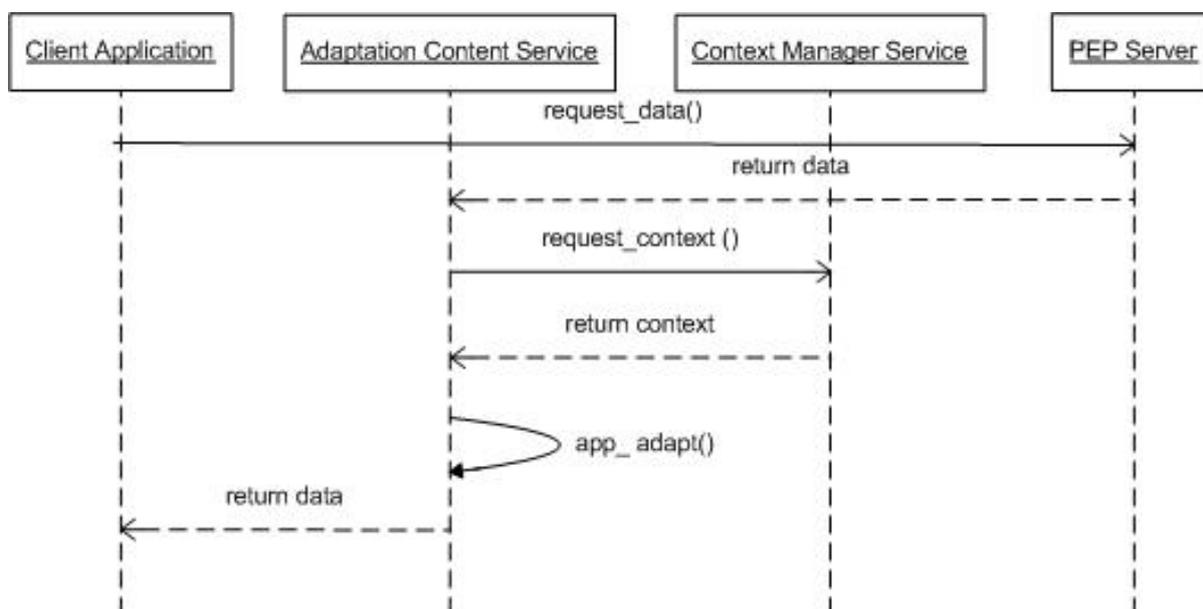


Figura 4.16 Diagrama de Seqüência - Adaptação de Conteúdo

A seqüência de passos para a adaptação de conteúdo obedece a seguinte ordem:

1. O médico solicita um determinado conteúdo a um servidor de dados (PEP) de um hospital credenciado ao ambiente UHS;

2. O serviço de adaptação de conteúdo solicita ao serviço de gerenciamento de contexto, o contexto do usuário-médico;
3. O serviço de gerenciamento de contexto retorna o contexto ao serviço de adaptação de conteúdo;
4. O serviço de adaptação de conteúdo seleciona uma política de adaptação;
5. Ao receber os dados do PEP, solicitados pelo médico, o serviço de adaptação de conteúdo aplica a política de adaptação aos dados enviados pelo PEP e só então, os envia ao médico.

Para o cenário UHS, o conteúdo é tratado no serviço de adaptação de conteúdo, disponibilizado pela arquitetura de *middleware* e serviços que dá suporte ao ambiente. Observa-se, então que, conforme apresentado no capítulo 2 e definido por Lei e Nicolas (Lei e Nicolas 2001), a adaptação ocorre no *middleware*. A parte referente aos dados é de responsabilidade do PEP, que de fato, contém os dados e as informações dos pacientes. Para selecionar o formato a ser enviado ao dispositivo do usuário, as informações de contexto serão repassadas pelo serviço de gerenciamento de contexto. O serviço de adaptação de conteúdo aplica a política da adaptação aos dados e os apresenta ao usuário requisitante.

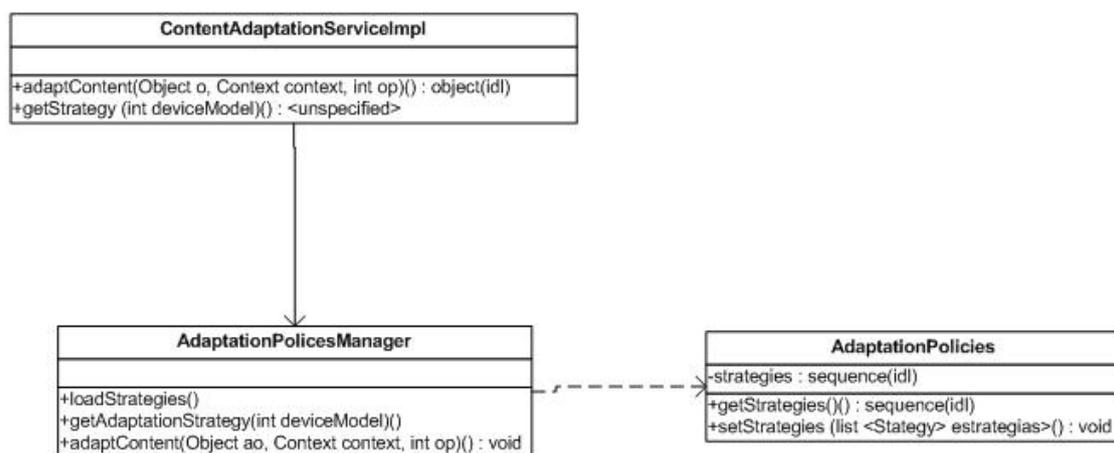


Figura 4.17 Diagrama de Classes do Serviço de Adaptação de Conteúdo

Como visto, o serviço de adaptação de conteúdo atua na adaptação à infra-estrutura técnica, citada por (Lei e Nicolas 2001). Neste caso, baseado no contexto do equipamento, cuja informação é repassada pelo serviço de gerenciamento de contexto, é disponibilizado um formato para apresentação do conteúdo ao usuário.

4.3.1 Políticas de Adaptação

As informações contidas nos PEPs contêm dados completos sobre os pacientes (antecedentes, histórico, casos clínicos, etc). Um dos problemas que o serviço de adaptação de conteúdo objetiva minimizar é a diversidade de configurações e tamanhos de telas dos dispositivos e suas possíveis limitações. Dessa forma, a depender do dispositivo utilizado pelo médico para acesso ao ambiente UHS, haverá ou não a filtragem de partes das informações contidas no prontuário eletrônico do paciente.

Um telefone celular, por exemplo, é um dispositivo muito utilizado para acesso ao ambiente, embora ofereça pouco conforto para a inclusão de informações em grande volume devido à sua interface e teclado limitados. Com base nisto, parte-se do princípio que o médico não irá realizar inclusões de casos clínicos fazendo uso de um dispositivo de interface limitada, tendo em vista que tais dados são incluídos no momento em que o paciente é atendido em uma unidade de saúde (hospital ou posto de saúde). Sendo assim, tal funcionalidade não precisa ser disponibilizada para os médicos que realizam acesso através do celular, por exemplo.

Por outro lado, as atividades mais prováveis de serem realizadas fazendo uso de um telefone celular são atividades de consulta a dados do PEP e solicitações e respostas a pareceres médicos em ações de segunda opinião. Baseando-se nisto, o ambiente UHS disponibiliza um módulo de acesso aos PEPs usando telefones celulares que fazem uso do serviço de adaptação de conteúdo para a realização de filtrações de informações a serem disponibilizadas ao médico, quando o mesmo realiza um acesso através de um dispositivo celular. O objetivo principal dessas filtrações de conteúdo é fazer com que informações relevantes sejam apresentadas ao usuário e informações menos críticas possam ser suprimidas para minimizar os problemas com limitações de telas e teclados de tais dispositivos.

O serviço de adaptação de conteúdo trabalha com 2 níveis de filtragem. A primeira refere-se à filtragem da quantidade de campos do PEP a serem exibidos no dispositivo de interface limitada. Já a segunda, refere-se à filtragem do conteúdo de cada campo a ser exibido.

Para realizar a primeira filtragem é utilizada a variável de contexto do dispositivo. Ao ser detectado um dispositivo de classe 3, algumas informações serão suprimidas. Este tipo de adaptação é chamado de Estratégia 1 para o serviço de adaptação de conteúdo do *UbiDoctor*.

Para realização da segunda filtragem, será utilizado o conceito de *stop list*. Uma *stop list*, ou dicionário negativo é um recurso usado para indexação automática na filtragem de palavras (Fox 1989).

Através do uso de *stop list*, um campo do tipo *string* pode ser regenerado com um número menor de caracteres através da eliminação de preposições, artigos, conjunções e conectivos,

eliminando, assim palavras menos relevantes do conteúdo da informação. Essa estratégia (Estratégia 2), por exemplo, pode reduzir as informações associadas ao histórico do paciente, hipótese médica, dentre outros.

O serviço de adaptação de conteúdo do *UbiDoctor* utiliza o nível de filtragem de *stop list* para eliminar palavras descartáveis com o objetivo de reduzir o número de caracteres de um campo de descrição do tipo *string* para a interface de dispositivos com tela de tamanho limitado.

Além da estratégia de *stop list*, o serviço de adaptação de conteúdo do *UbiDoctor* utiliza uma filtragem baseada em palavras-chave (Estratégia 3). Esta estratégia é utilizada quando a estratégia de *stop list* não gera como saída uma *string* com tamanho inferior ao suportado pelos campos do tipo *string* da interface dos telefones celulares.

O repositório de palavras-chave contém um dicionário de dados com termos médicos referentes a doenças, tratamentos, medicamentos e palavras que constantemente aparecem em PEP e permite que novas palavras possam ser cadastradas.

Justifica-se a separação das estratégias 2 e 3 pelo desempenho da tarefa de adaptação de conteúdo. Caso fosse realizada a estratégia 3, sem a realização da filtragem por *stop list*, o número de palavras a serem buscadas no repositório tenderia a ser maior e demandaria mais tempo na busca por palavras irrelevantes, fazendo com que o tempo gasto no processo de adaptação fosse maior. Uma vez que menos palavras serão submetidas ao repositório de palavras-chave, menos tempo será gasto com a estratégia 3 e conseqüentemente com o processo de adaptação de conteúdo. É importante ressaltar a preocupação com o tempo gasto no processo de migração. Mesmo mediante situações onde se faça necessária a adaptação de conteúdo, o tempo total decorrido deverá ser pequeno, de modo a representar um baixo impacto no tempo observado entre duas atividades executadas sucessivamente. O uso dessa arquitetura poderá motivar o médico a agir de maneira mais rápida possibilitando o aumento de sua produtividade.

A estratégia baseada em palavras-chave utiliza um repositório de palavras que será consultado e toda palavra encontrada, na *string* de saída após a estratégia da *stop list*, será procurada neste repositório. Caso a palavra não seja encontrada no repositório, a mesma será eliminada do campo do conteúdo. Isto será feito com todas as palavras contidas na *string* que representa o campo após a aplicação da estratégia de *stop list*. Ao término do processo, o campo a ser mostrado na tela contém apenas palavras relevantes para a análise dos casos clínicos.

Para os casos onde, mesmo após a aplicação das 3 estratégias, ainda há mais caracteres que o comportado pelo campo, será incluído um novo campo de dados na tela do dispositivo do médico (Estratégia 4). Este campo refere-se à continuação do anterior, para comportar todos os dados a serem apresentados após o processo de filtragem de dados. O esquema apresentado na Figura 4.18 mostra as estratégias aplicadas para a adaptação do conteúdo, utilizadas pelo

serviço do *UbiDoctor*.

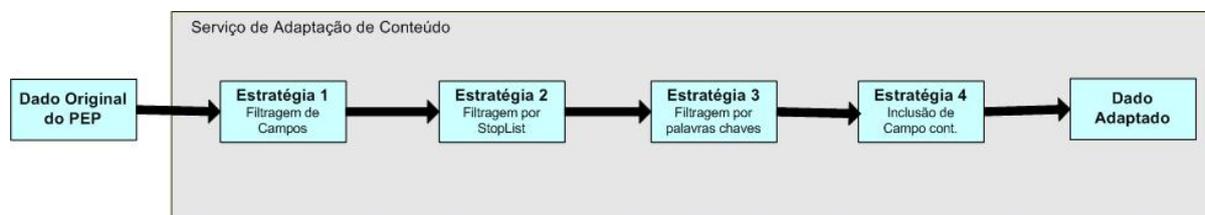


Figura 4.18 Aplicação das Estratégias de Adaptação

A combinação das estratégias irá definir a política da adaptação. Caso, após a aplicação da estratégia 1, todos os campos de dados a serem mostrados apresentem menos caracteres que o limite máximo permitido, não será necessário a aplicação das outras estratégias. Caso contrário, outras estratégias precisarão ser aplicadas para que o conteúdo possa ser adaptado e mostrado na tela do dispositivo móvel. Ao término da aplicação das estratégias, ou seja, escolha e aplicação da política de adaptação, o dado adaptado será mostrado na tela do dispositivo do usuário.

4.3.2 Migração e Adaptação de Conteúdo

Para o ambiente UHS, a migração e a adaptação de conteúdo tem uma relação estreita, uma vez que ambos estão relacionados à questão de multiplataforma. Como foi visto na seção anterior, existem dois tipos de migração considerados pelo presente trabalho: *Hardware-driven* e *User-driven*.

Para o caso do *hardware-driven*, o aviso parte do dispositivo e após a confirmação e escolha do usuário com relação ao novo dispositivo a ser utilizado, o contexto novo será inspecionado. Ao mesmo tempo, o serviço de adaptação de conteúdo é avisado da mudança de dispositivo e para isto, o conteúdo deve ser adaptado ao novo contexto. Uma nova consulta é submetida ao servidor de dados da aplicação e este, por sua vez, re-envia os dados ao serviço de adaptação de conteúdo que, por fim, os re-estrutura baseando-se em uma política e no contexto do dispositivo cliente. O diagrama de seqüência que apresenta o conjunto de tarefas do serviço de adaptação para prover a migração *hardware-driven* pode ser observado na Figura 4.19.

Quando a migração ocorre por iniciativa do usuário, o que foi denominado pelo presente trabalho *user-driven*, ele escolhe para qual dispositivo deseja transferir a sessão. Uma vez escolhido, o serviço de gerenciamento de sessão conduzirá o processo. O serviço de adaptação de conteúdo é avisado da mudança e após a obtenção do novo contexto, a nova política de adaptação poderá ser gerada e aplicada aos dados.

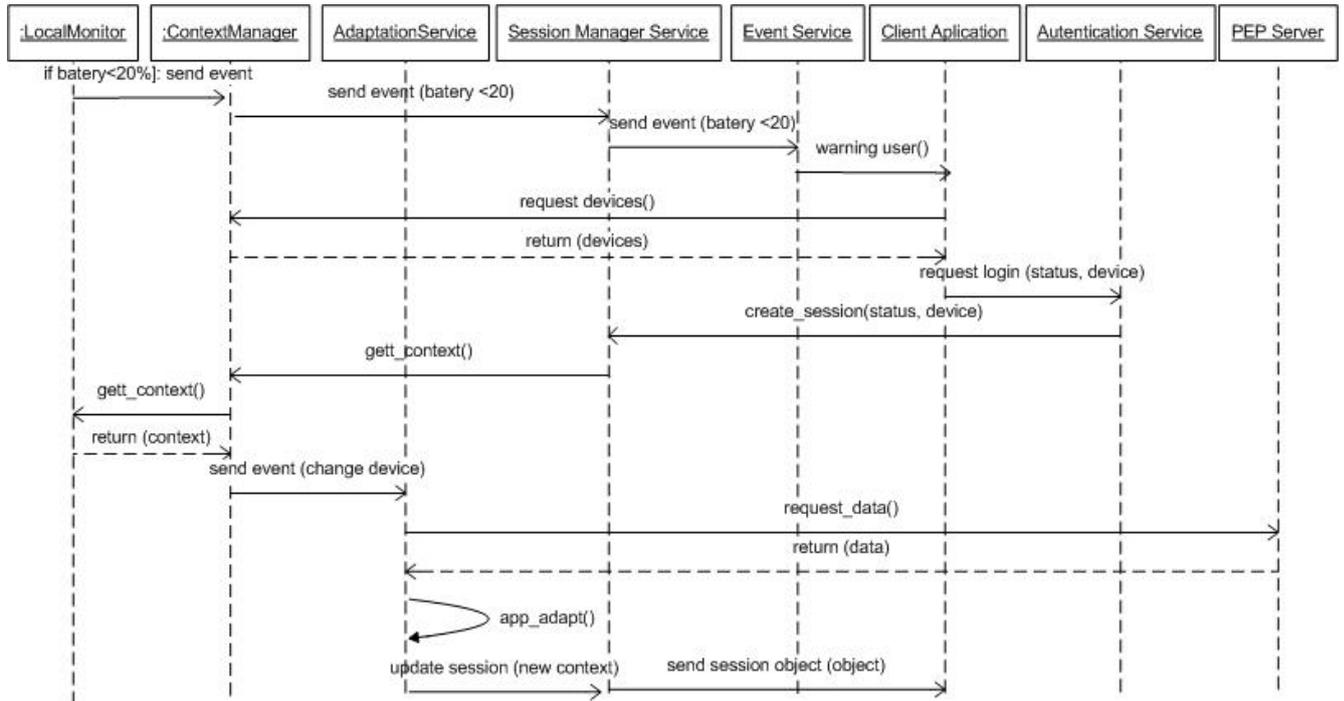


Figura 4.19 Diagrama de Seqüência - Migração de Sessão *Hardware-driven* e Adaptação

4.4 Considerações Finais

O presente trabalho apresenta a arquitetura do *middleware UbiDoctor*. Esta arquitetura é composta por 3 serviços que também foram definidos neste capítulo. Foi definido um serviço que gerencia as sessões de aplicações, controlando persistência, interrupções e migração de sessões, e um serviço que possibilita a realização de ações de adaptação de conteúdo baseadas em variáveis de contexto, tendo em vista a variedade de dispositivos envolvidos no cenário. É importante ressaltar que a arquitetura proposta é baseada em serviços comuns de *middleware*, estando pois, desacoplada da aplicação, podendo então ser re-utilizada em cenários que apresentem requisitos similares aos do UHS.

Os serviços de gerenciamento de sessão e de adaptação de conteúdo se propõem a minimizar problemas referentes a atrasos e interrupções no processo de migração de sessões, objetivando ganhos de produtividade na rotina de trabalho de profissionais que desempenham atividades ricas em fragmentação e nomadismo. Ambos os serviços utilizam o serviço de gerenciamento de contexto como suporte às suas atividades, que também foi apresentado no presente capítulo.

Para avaliar se a possibilidade destes ganhos de produtividade de fato pôde ser obtida, foi construído um protótipo de uma aplicação de PEP com segunda opinião médica que faz uso da

infra-estrutura de serviços do *UbiDoctor*, que está apresentado no capítulo 5. O capítulo 6, por sua vez, apresentará os experimentos realizados no cenário UHS, bem como as impressões dos médicos a cerca do que foi proposto.

Prototipação

Para avaliar os serviços propostos pela arquitetura *UbiDoctor*, apresentada no capítulo anterior, foi necessário o desenvolvimento de uma aplicação protótipo que faz uso de tais serviços.

Este capítulo apresenta a implementação da arquitetura de serviços de *middleware UbiDoctor* 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, através do qual médicos têm acesso, a qualquer hora, de qualquer lugar e usando qualquer dispositivo, a dados de pacientes contidos em PEPs distribuídos nos hospitais e postos de saúde. A aplicação ainda permite que o médico 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.

5.1 Arquitetura do Protótipo

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 re-usados em outros ambientes com características e requisitos similares, já que aparecem como serviços comuns de *middleware*.

Os serviços de gerenciamento de contexto, gerenciamento de sessão e adaptação de conteúdo foram implementados conforme definidos na arquitetura apresentada no Capítulo 4. A implementação dos serviços foi realizada usando a linguagem de programação Java, através do ambiente de desenvolvimento NetBeans IDE 5.5, e infra-estrutura de comunicação baseada em *sockets* e *Java-Remote Method Invocation* (RMI). A escolha desta linguagem se deu devido à portabilidade, recursos de rede, grande quantidade de APIs e sua adequação para dispositivos como PDAs e telefones celulares.

Para testar a utilização dos serviços, foi necessário o desenvolvimento de uma aplicação protótipo, onde os médicos atuam nas suas atividades diárias dentro da sua perspectiva nômade. O aplicativo desenvolvido foi chamado de UHSys que utilizou as bibliotecas dos serviços de *middleware UbiDoctor* para facilitar o seu desenvolvimento. Além disto, o *UbiDoctor* favo-

rece a realização de algumas ações dentro do sistema através do suporte dado, em tempo de execução, pelos seus serviços.

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 possível que o médico faça solicitações de segunda opinião médica (solicitação de parecer) e analise possíveis solicitações de pareceres enviadas a ele.

No protótipo não foram considerados problemas relativos à heterogeneidade da informação que está representada nas diversas bases de dados nos PEPs. Embora tenha-se conhecimento da dificuldade de integração da informação, este problema está fora do escopo desta tese. 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. Essa migração deve ser realizada com garantias de persistência nos dados e ainda, no menor intervalo tempo possível, possibilitando 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. Existe ainda a necessidade de adaptar o conteúdo a ser mostrado ao médico, a depender 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 arquitetura do UHSys pode ser observada na Figura 5.1. 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, que no caso específico da prototipação deste trabalho, foi utilizado o Apache TomCat. 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 clientes.

No *front-end* do sistema, existem clientes que acessam o ambiente através de telefones celulares, fazendo uso de uma interface desenvolvida em JME (*Java Micro Edition*), ou clientes que utilizam a interface *web-browser* de seus computadores pessoais e PDAs. Através do *front-end*, um médico poderá consultar um registro de um prontuário eletrônico de um paciente e acrescentar informações demográficas, clínicas, exames, prescrições, dentre outros dados. Os dados do paciente podem estar em prontuários eletrônicos 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 conjunto de dispositivos variados e um conjunto de possibilidades de redes de acesso.

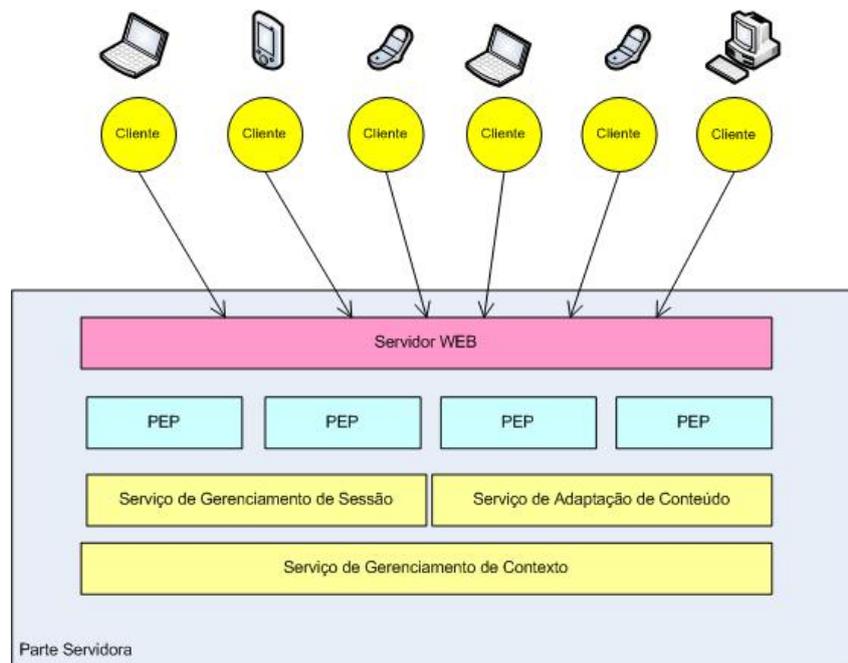


Figura 5.1 Arquitetura do Protótipo

5.2 Implementação dos Serviços

O *UbiDoctor* foi implementado em três bibliotecas principais: *services.context*, *services.session*, *services.ContentAdaptation*. A primeira delas refere-se ao serviço de gerenciamento de contexto, a segunda, ao de gerenciamento de sessão e a última, ao serviço de adaptação de conteúdo. Além deste conjunto de pacotes, outras classes de apoio também são utilizadas como por exemplo, *services.device*, *services.location* e *services.util*.

O pacote *services.device* pode ser visualizado na Figura 5.2. Observa-se que ele contém a classe *Device* que representa o dispositivo, a classe *DeviceRepository* que é o repositório com todos os dispositivos cadastrados no ambiente, inclusive com as suas respectivas localizações e a classe *DeviceManager* que se refere à classe que manipula os dados do repositório de dispositivos. Por fim, a classe *LocalMonitor* representa o monitor local que fica instalado no dispositivo e irá se comunicar com o gerenciador de contexto do serviço de gerenciamento de contextos.

O pacote *services.location* contém as classes referentes ao repositório de localizações e a classe *Location* que representa informações sobre localização simbólica e relativa do dispositivo. O pacote *services.util* por sua vez apresenta classes utilitárias para a comunicação no nível mais básico.

Após o processo de compilação das classes do serviço, são gerados dois arquivos com ex-



Figura 5.2 Pacote services.device

tensão jar, para serem importados por qualquer aplicativo que venha a utilizar o *UbiDoctor* como *middleware*, para facilitar a construção de aplicações. As bibliotecas geradas são *services.jar* e *physician.jar*. A primeira delas refere-se a todas as classes citadas previamente nesta sessão. A biblioteca *physician.jar* representa a classe associada a características do médico. As subseções a seguir detalham a implementação dos serviços do *middleware UbiDoctor*.

5.2.1 Serviço de Gerenciamento de Contexto

O Serviço de Gerenciamento de Contexto utiliza uma comunicação cliente/servidor baseada em *socket*, onde um cliente de contexto comunica-se com o servidor para informar possíveis mudanças de contexto. O serviço de contexto utiliza bibliotecas do *framework* MoCA. Um monitor do MoCA executa nos clientes móveis do tipo *notebook* e coleta informações sobre localização (o endereço MAC do *AP-Access point* que o dispositivo utiliza para se conectar à rede), memória e CPU utilizáveis e bateria do dispositivo. Outras informações com relação a endereços IP local e dos APs também são disponibilizados pelo monitor do MoCA. O cliente do serviço de contexto possui um monitor local do *UbiDoctor* (*LocalMonitor*) que se comunica com o *ContextManagerServer* para repassar dados iniciais sobre o contexto do médico no momento do login e também durante a execução da sessão, repassando informações de mudança de contexto. A comunicação assíncrona utiliza o serviço de eventos do MoCA (*ECIServer*).

Para a inicialização do serviço de contexto, será necessário a execução do método *init* que é chamado pelo método *main* do *ContextManagerServer*. Ao se executar o *ContextManagerServer*, o serviço de contexto entrará no ar. O código fonte do método *init* pode ser visto na Figura 5.3, no quadro mais à esquerda. No quadro da direita, da Figura 5.3, são apresentadas as mensagens emitidas na inicialização do serviço de contexto.

```

public void init() {
    ThreadPool pool = new ThreadPool(25);
    pool.startPool();
    ServerSocket serverSocket = null;
    try {
        serverSocket = new
ServerSocket(ContextGlobals.SERVICE_PORT);
    } catch (IOException e) {
        System.out.println("Port Already busy... Quitting Service...");
        this.log.error("Port Already busy... Quitting Service...");
        System.exit(0);
    }
    //conexao via tcp/udp
    this.acceptor = new ContextConnectionAcceptor(serverSocket, pool,
this);
    this.acceptor.start();
    System.out.println("Acceptor running at Port:" +
this.acceptor.getPort());
    this.log.info("Acceptor running at Port:" + this.acceptor.getPort());
    //conexoes via http
    ContextHttpServer server = ne
ContextHttpServer(ContextGlobals.HTTP_SERVICE_PORT, pool, this);
    server.start();
    System.out.println("HTTPAcceptor running at Port:" + server.getPort());
    this.log.info("HTTPAcceptor running at Port:" + server.getPort());
    this.log.info("Service running...");
    System.out.println("Service running...");
}
}

```

```

Acceptor running at Port:9001
0 [main]
INFO services.context.ContextService -
Acceptor running at Port:9001

HTTPAcceptor running at Port:9002
47 [main] INFO
services.context.ContextService -

HTTPAcceptor running at Port:9002
47 [main] INFO
services.context.ContextService - Service
running...
Service running...
WorkerThread_0: executed

```

Figura 5.3 Inicialização do Serviço de Gerenciamento de Contexto

5.2.2 Serviço de Gerenciamento de Sessão

O Serviço de Gerenciamento de Sessão utiliza uma arquitetura baseada em objetos distribuídos implementada através de Java-RMI. Com Java-RMI, os objetos podem ser localizados, comunicam-se através de um protocolo padrão e colaboram entre si. Um cliente submete uma tarefa a ser executada pelo objeto remoto (também chamado de objeto servidor) que executa a tarefa e devolve o resultado da execução para o cliente que o invocou. Para o serviço de Gerenciamento de Sessão foram definidos os seus métodos juntamente com os parâmetros de entrada e retorno, compondo assim, a interface do serviço, aqui denominada *SessionManager*. Através da interface é que o cliente saberá quais parâmetros devem ser passados ao serviço. Toda interface deverá estender a interface *Remote* padrão do Java-RMI. O *SessionManagerImpl*, por sua vez, é um objeto remoto implementando a interface *SessionManager* e estende a classe *UnicastRemoteObject*.

O código fonte que apresenta a interface *SessionManager* e os seus métodos pode ser observado na Figura 5.4.

O *sessionManager* possui uma lista de sessões ativas e cada sessão apresenta um ou mais médicos participantes, e cada médico tem o seu contexto. Para que o serviço se torne aces-

```
package services.session;

import java.rmi.*;

import physician.*;

public interface SessionManager extends Remote{

    public void create_session(String u, int code, int user_id, String referencia, int
cod_referencia, int status_sessao, int tipoTela) throws RemoteException;
    public void update_session(int userId, String login, String referencia, int
cod_referencia, int status_sessao, int tipoTela) throws RemoteException;
    public void remove_session(String ID) throws RemoteException;

    public void send_event() throws RemoteException;

    public Session query_session(String ID) throws RemoteException;

    public void application_roaming(String login, String url) throws RemoteException;

    public int getStatus() throws RemoteException;

    public SessaoAplicacao query_sessao_aplicacao(int userId) throws
RemoteException;

    public SessaoAplicacao query_sessao_aplicacao(String userLogin) throws
RemoteException;

    public void update_session(Session session, SessaoAplicacao sessaoAplicacao)
throws RemoteException;
}
```

Figura 5.4 Interface do Serviço de Gerenciamento de Sessão

sível, é necessário que um programa registre-o no serviço de nomes (*rmiregistry*), iniciando a sua execução. Para o serviço de Gerenciamento de Sessão, este programa é denominado *SessionManagerServer* e o seu código fonte pode ser visualizado na Figura 5.5.

```
import java.rmi.registry.*;

public class SessionManagerServer {

    public SessionManagerServer(){
        runServer();
    }

    public void runServer(){

        try{

            Registry r = LocateRegistry.createRegistry(Registry.REGISTRY_PORT);
            SessionManagerImpl server = new SessionManagerImpl();
            r.rebind("SessionManagerServer",server);

        } catch(Exception e){
            System.out.println("Trouble: " + e );
        }
    }

    public static void main(String[] args) throws Exception{
        System.err.println("Initializing server: please wait.");

        new SessionManagerServer();

        System.err.println("The Session Manager server is up and running.");
    }
}
```

Figura 5.5 Inicialização do Serviço de Gerenciamento de Sessão

Para cada cliente, uma instância da classe *SessionManagerLocal* é criada, representando um cliente Java-RMI para o serviço de gerenciamento de sessões. A classe *SessionManagerLocal* é composta de um único método denominado *runClient* cujo tipo de retorno é a interface *SessionManager*, através da qual é realizado o acesso ao objeto remoto que representa a implementação do serviço de gerenciamento de sessão (*SessionManagerImpl*).

5.2.3 Serviço de Adaptação de Conteúdo

O serviço de Adaptação de conteúdo também foi modelado com objetos e usa Java-RMI. O pacote *services.ContentAdaptation* do *UbiDoctor* refere-se a todas as classes que tratam da adap-

tação de conteúdo. O *ContentAdaptationService* é a interface que estende a interface padrão do Java-RMI(*Remote*) e *ContentAdaptationServiceImpl* contém a implementação dos métodos *adaptContentString* e *getStrategy* que se refere à ação de adaptar o conteúdo e adquirir a estratégia de adaptação a ser utilizada. O código fonte que representa a implementação do serviço de adaptação de conteúdo pode ser observado da Figura 5.6.

```
package services.contentAdaptation;

import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

import org.apache.log4j.Logger;

import services.context.Context;

public class ContentAdaptationServiceImpl extends UnicastRemoteObject
implements ContentAdaptationService {

    private AdaptationStrategyManager strategyManager = null;

    public String adaptContentString(String content, Context context, int op) {
        System.out.println("Adapting Called...");
        return this.strategyManager.adaptContentString(content, context, op);
    }

    public AdaptationStrategy getStrategy(int deviceModel) throws
RemoteException {
        AdaptationStrategy strategy =
            this.strategyManager.getAdaptationStrategy(deviceModel);
        return strategy;
    }
}
```

Figura 5.6 Serviço de Adaptação de Conteúdo

Foi definido um gerenciador de estratégia de adaptação que implementa métodos para o tratamento das informações do tipo *string*, aplicando as estratégias de adaptação conforme citadas no Capítulo 4. Para realizar a adaptação, o serviço utiliza um arquivo de apoio que contém a lista de palavras que podem ser excluídas do dado quando o mesmo vai ser apresentado em um dispositivo de tela pequena. Este processo de tratamento do dado utiliza a estratégia de *StopList* e o seu código pode ser visualizado na Figura 5.7.

```

public class Estrategia_StopList {

private StopList stopList;

public Estrategia_StopList(StopList stopList) {
    this.stopList = stopList;}
    public String filtrar(String texto) {
        int indiceAux;
        int tamanhoPalavra;
        StringBuffer strTemp = new StringBuffer();
        Vector<String> listaPalavras = stopList.getListaPalavras();
        Vector<String> listaCaracteres = stopList.getListaCaracteres();
        StringBuffer conteudo = new StringBuffer(texto);
        // Removendo os caracteres
        for (int i=0 ; i<listaCaracteres.size() ; i++) {
            indiceAux = conteudo.toString().toUpperCase().indexOf(listaCaracteres.elementAt(i).toUpperCase());
            if (indiceAux >= 0) {
                conteudo.deleteCharAt(indiceAux);
                indiceAux = conteudo.toString().toUpperCase().indexOf(listaCaracteres.elementAt(i).toUpperCase());
            }
        }
        // Removendo as palavras
        for (int i=0 ; i<listaPalavras.size() ; i++) {
        // Removendo a palavra do meio do texto
            strTemp = new StringBuffer();
            strTemp.append(" ");
            strTemp.append(listaPalavras.elementAt(i).toUpperCase());
            strTemp.append(" ");
            indiceAux = conteudo.toString().toUpperCase().indexOf(strTemp.toString());
            tamanhoPalavra = strTemp.toString().length();
            while (indiceAux >= 0) {
                conteudo.delete(indiceAux, indiceAux+tamanhoPalavra-1);
                indiceAux = conteudo.toString().toUpperCase().indexOf(strTemp.toString());
            }
        // Removendo a palavra do começo do texto
            strTemp = new StringBuffer();
            strTemp.append(listaPalavras.elementAt(i).toUpperCase());
            strTemp.append(" ");
            if (conteudo.toString().toUpperCase().startsWith(strTemp.toString())) {
                conteudo.delete(0, strTemp.toString().length());
            }
        // Removendo a palavra do fim do texto
            strTemp = new StringBuffer();
            strTemp.append(" ");
            strTemp.append(listaPalavras.elementAt(i).toUpperCase());
            if (conteudo.toString().toUpperCase().endsWith(strTemp.toString())) {
                conteudo.delete(conteudo.toString().length()-strTemp.toString().length(), conteudo.toString().length());
            }
        }return conteudo.toString();
    }
}

```

Figura 5.7 Implementação da StopList

5.3 Implementação do Servidor de Aplicação

A implementação do servidor de aplicação tomou como base as informações utilizadas pelo projeto *HealthNet*¹, desenvolvido pelo grupo TIS do núcleo de Telesaúde sediado no Hospital das Clínicas da Universidade Federal de Pernambuco. Optou-se pelo uso de tais informações devido ao fato da obtenção dos requisitos para a construção deste ambiente (*HealthNet*) ter contado com médicos e profissionais da área de tecnologia da informação, o que fez com que o ambiente se aproximasse o máximo possível do ambiente real de trabalho.

No protótipo foi implementado um banco de dados com os prontuários eletrônicos de pacientes, utilizando o SGBD (Sistema de Gerenciamento de Banco de Dados) PostgreSQL 8.1. O modelo de dados do servidor de aplicação do PEP está apresentado na Figura 5.8, enquanto o diagrama de classes da aplicação *UHSys* pode ser observado na Figura 5.9. Este último diagrama apresenta todas as entidades presentes no aplicativo, que se refere ao servidor de aplicação de PEPs.

5.4 Implementação do Cliente

Para implementar os acessos de médicos através de dispositivos móveis foi desenvolvido um ambiente *UHSys* usando *Java Micro Edition* (JME). Foi utilizado o emulador *Sun Java™ Wireless Toolkit for CLDC*, versão 2.5.1, para simular a interface celular. Através deste ambiente, médicos poderão acessar os dados de pacientes dos prontuários eletrônicos distribuídos na rede UHS para realizar uma análise ou diagnóstico. Ainda é possível gerar respostas às solicitações de segunda opinião requisitadas por outros médicos, bem como analisar casos clínicos.

Além desta aplicação-cliente, também é possível fazer acesso ao sistema utilizando uma interface baseada em *web-browsers*. O *UHSys* para prover acesso através de dispositivos como *notebooks* e *desktops* disponibiliza páginas em JSP e Html. A aplicação cliente para Web é bem mais completa que a disponibilizada através dos celulares, devido aos recursos disponíveis em tais plataformas. Pelo fato de se utilizar telas e teclados maiores, é possível se obter todos os dados armazenados no PEP, a exemplo dos casos clínicos de cada paciente, dados pessoais e familiares, pareceres médicos anteriores, solicitar segunda opinião de um caso clínico a colegas médicos, dentre outras atividades. A página inicial da aplicação-cliente com acesso através de um *web-browser* pode ser observada na Figura 5.10.

Na Figura 5.10 observa-se, na parte superior, o menu principal com as funcionalidades da

¹ www.nutes.ufpe.br/healthnet

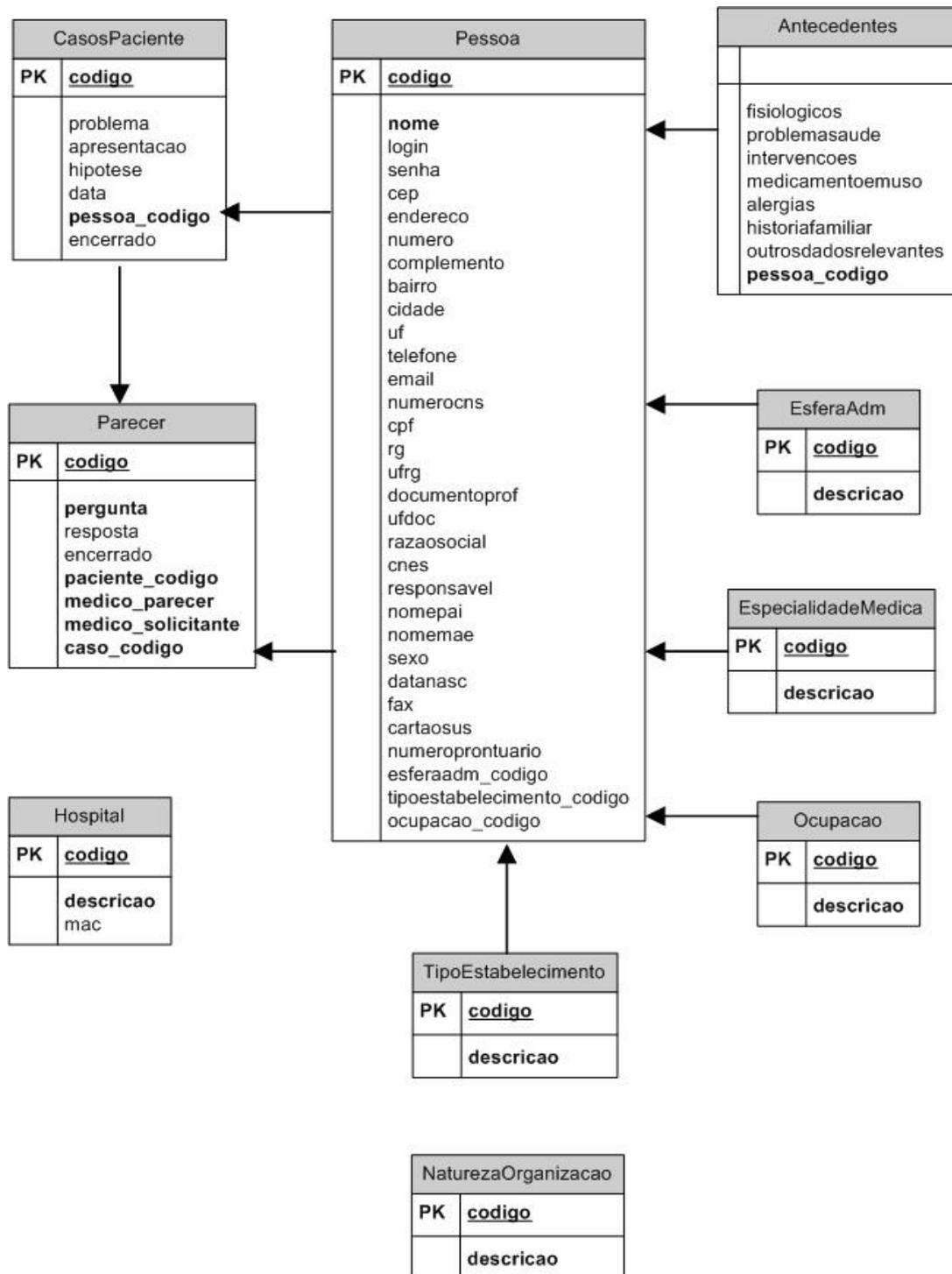


Figura 5.8 Entidades do Banco de Dados

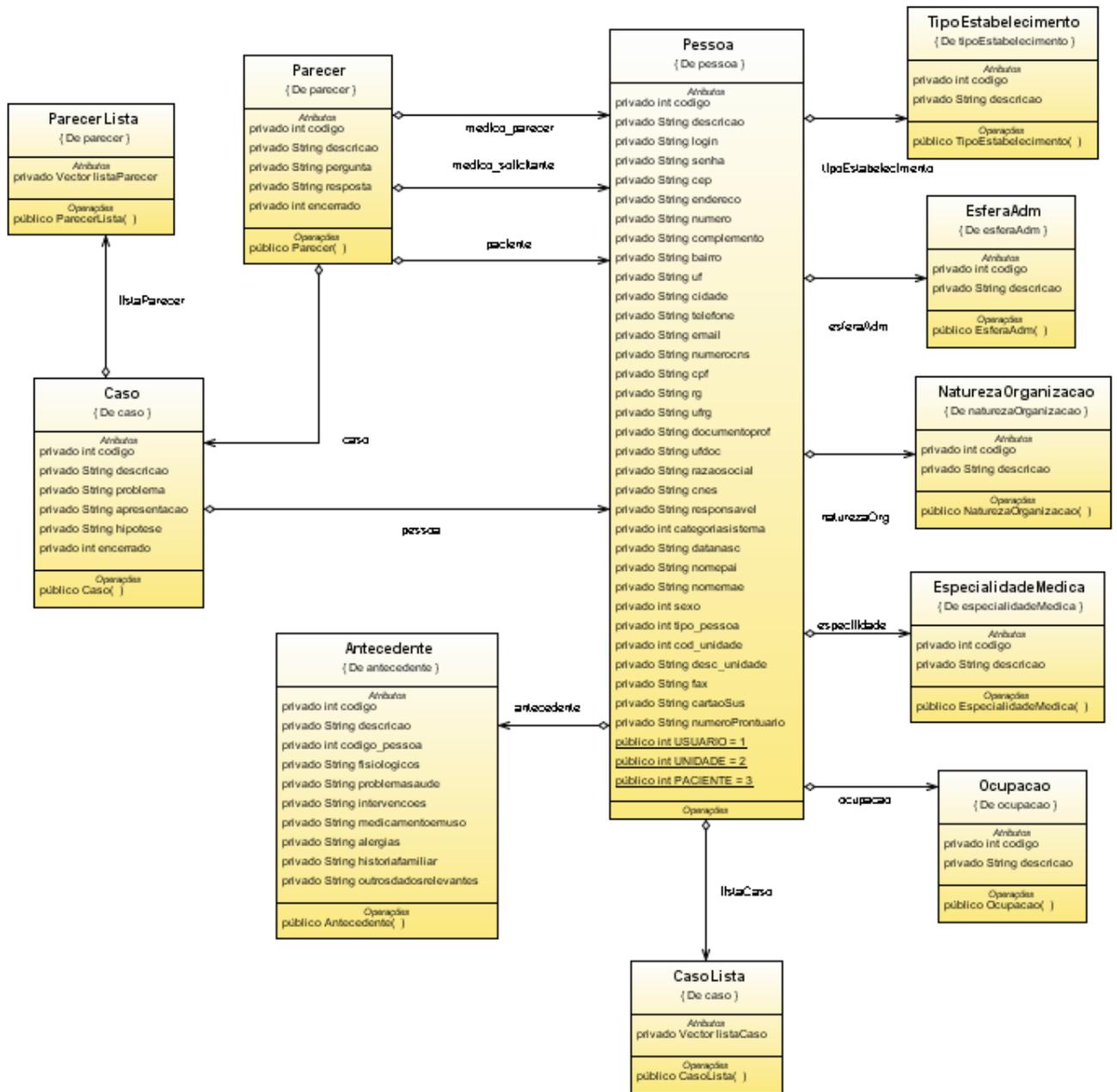


Figura 5.9 Diagrama de Classes do Servidor de Aplicação

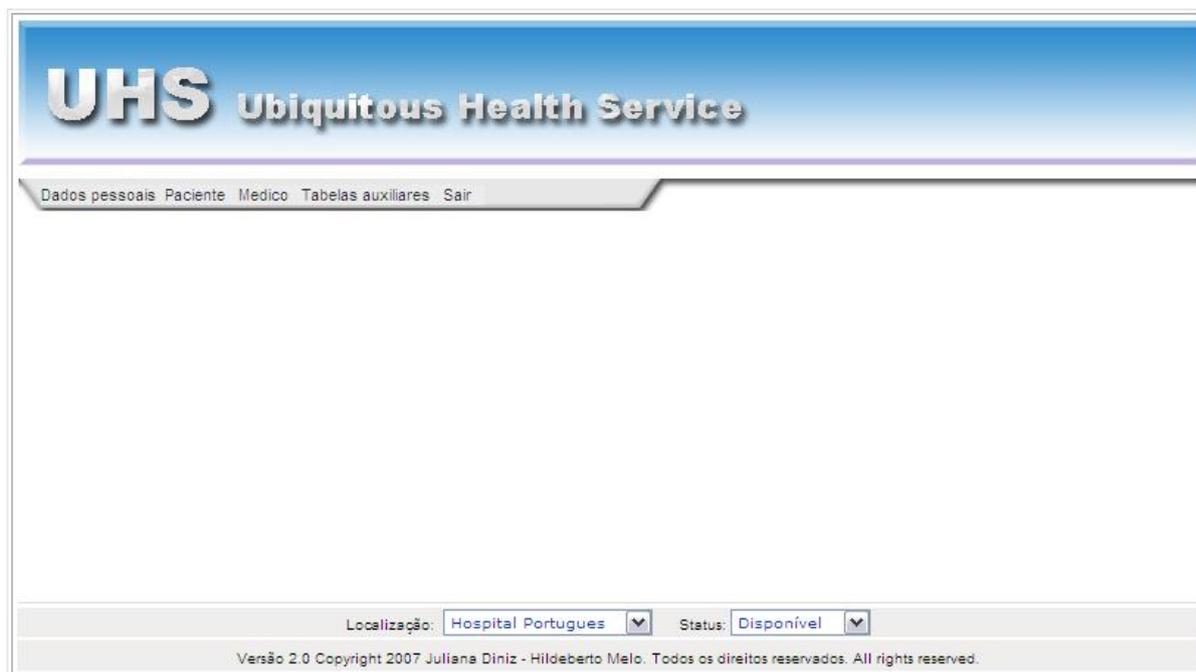


Figura 5.10 Tela Principal de acesso via Web Browser

aplicação. São os dados do médico naquela sessão, as informações referentes aos pacientes, informações referentes a outros médicos cadastrados no ambiente e, por fim, as tabelas auxiliares, que se referem às unidades de saúde e à esfera administrativa do ambiente. Na parte inferior da página principal encontram-se duas informações de contexto: status e localização do médico. Tais informações são repassadas para a aplicação cliente através do serviço de gerenciamento de contexto, mas se apresentam no modo de combo-box na interface para permitir que o médico modifique tais opções. Por exemplo, caso o médico não deseje que a sua localização seja exibida no sistema, para preservar sua privacidade, ele poderá trocar a localização sinalizada pela localização "indeterminada". Da mesma forma, caso o médico não aceite o status sugerido pelo sistema, ele poderá alterá-lo.

Como apresentado no Capítulo 4, a decisão sobre a localização do usuário é obtida baseada na posição do dispositivo que o médico utiliza para a conexão. Caso ele esteja numa estação fixa, a localização da estação está previamente definida no repositório de dispositivos. Caso ele utilize uma rede WI-FI conectando-se a um AP cadastrado no repositório de localidades, essa localização também poderá ser inferida. No caso do usuário conectar-se através de um telefone celular, será admitido, para este protótipo, que a sua localização é tida como "indeterminada".

Sabe-se que existem outras estratégias para, uma vez que se utilize uma rede GPRS, descobrir a localização do usuário com relação a sua célula. Entretanto, estes casos não foram

tratados neste protótipo.

Para a determinação do status do médico é utilizada a variável de contexto de status, que sugere o status baseado na localização. Quando o médico encontra-se em locais tidos como indeterminados, parte-se do pressuposto que ele está em ambientes não cadastrados, ou seja, não está em casa, hospitais ou consultórios, havendo grande probabilidade de ele estar "ausente". Por exemplo, uma conexão através de um celular em um automóvel é tida como status "ausente", pois o usuário está no carro. Caso o usuário esteja num taxi, ou seja, não está dirigindo, ele pode selecionar o seu status como "disponível". Outra situação para o status ser tido ou inferido como "ausente" é aquela onde o usuário está dentro de um hospital, mas em áreas tidas como críticas, como, por exemplo, bloco cirúrgico. Para isto é utilizada a localização simbólica que identifica a que AP o usuário se conecta e infere a sua posição em termos de locais pré-definidos no banco de dados do LIS, serviço de inferência do MoCA, citado no Capítulo 4.

5.4.1 Interface Cliente para Web-Browsers

Ao utilizar um acesso através de um *web-browser*, as seguintes atividades do protótipo serão disponibilizadas ao médico:

- Alterar seus dados pessoais;
- Incluir um novo paciente;
- Consultar os dados de um paciente cadastrado;
- Acrescentar informações aos dados demográficos de um paciente;
- Acrescentar informações dos antecedentes de um paciente;
- Acrescentar um novo caso clínico a um paciente;
- Consultar um caso clínico específico;
- Acrescentar informações a um caso clínico já existente;
- Solicitar uma segunda opinião (parecer) de um caso clínico;
- Consultar pareceres de um caso clínico;
- Responder os pareceres (segundas opiniões) que lhes forem solicitados;
- Visualizar o cadastro de médicos do sistema;

- Visualizar o cadastro de hospitais e unidades de saúde.

Para alterar os seus dados, o médico deverá escolher a opção "Dados Pessoais" no menu da tela principal. Para consultar o PEP de algum paciente, deverá ser escolhida opção "Paciente" no menu. Este ação possibilita que seja incluído um novo paciente (tela apresentada na Figura 5.11) ou que seja realizada a consulta aos dados de um paciente previamente cadastrado no sistema.

The screenshot shows a web browser window titled "UHS - Cadastro Paciente - Windows Internet Explorer". The address bar shows the URL: `http://localhost:8084/UHS_web/sistema/cadastros/paciente/cadPaciente.jsp?codigoPessoaLogada=1&sessaoCodigo=2&array=[object]`. The page title is "CADASTRO DE PACIENTE - (Campos com * são obrigatórios)".

The form is divided into several sections:

- Personal Information:** Fields for Nome (*), Nascimento (*), Sexo (dropdown with 'M' selected), Nome do pai, Nome da mãe (*), Endereço (*), Complemento, Número (*), Bairro (*), Cidade (*), C.E.P., U.F. (dropdown with 'PE' selected), Telefone, E-mail (* Apenas um), and Unidade de origem (dropdown with "--UNIDADESAUDE--" selected).
- Documentos de identificação:** Fields for Número do prontuário, Cartão SUS, CPF, and RG (with U.F. dropdown set to 'PE').

At the bottom of the form, there are three buttons: "Salvar", "Migrar sessão", and "Fechar".

Figura 5.11 Tela para Incluir Novo Paciente

O conjunto de telas que visualiza a consulta de dados de um paciente já cadastrado pode ser observado na Figura 5.12. Neste caso, foi escolhida a opção Consulta no menu de Pacientes e a primeira tela apareceu. Em seguida, o médico forneceu o nome do paciente cujo PEP deveria ser consultado e a segunda tela foi apresentada, trazendo os dados do mesmo. Nesta segunda tela, são apresentados os casos clínicos associados àquele paciente e os pareceres já solicitados e respondidos, referentes ao caso. O médico poderá alterar o presente caso ou dar o seu parecer, se outro médico o solicitou. Caso não tenha sido solicitado um parecer àquele caso clínico, o botão estará desabilitado.

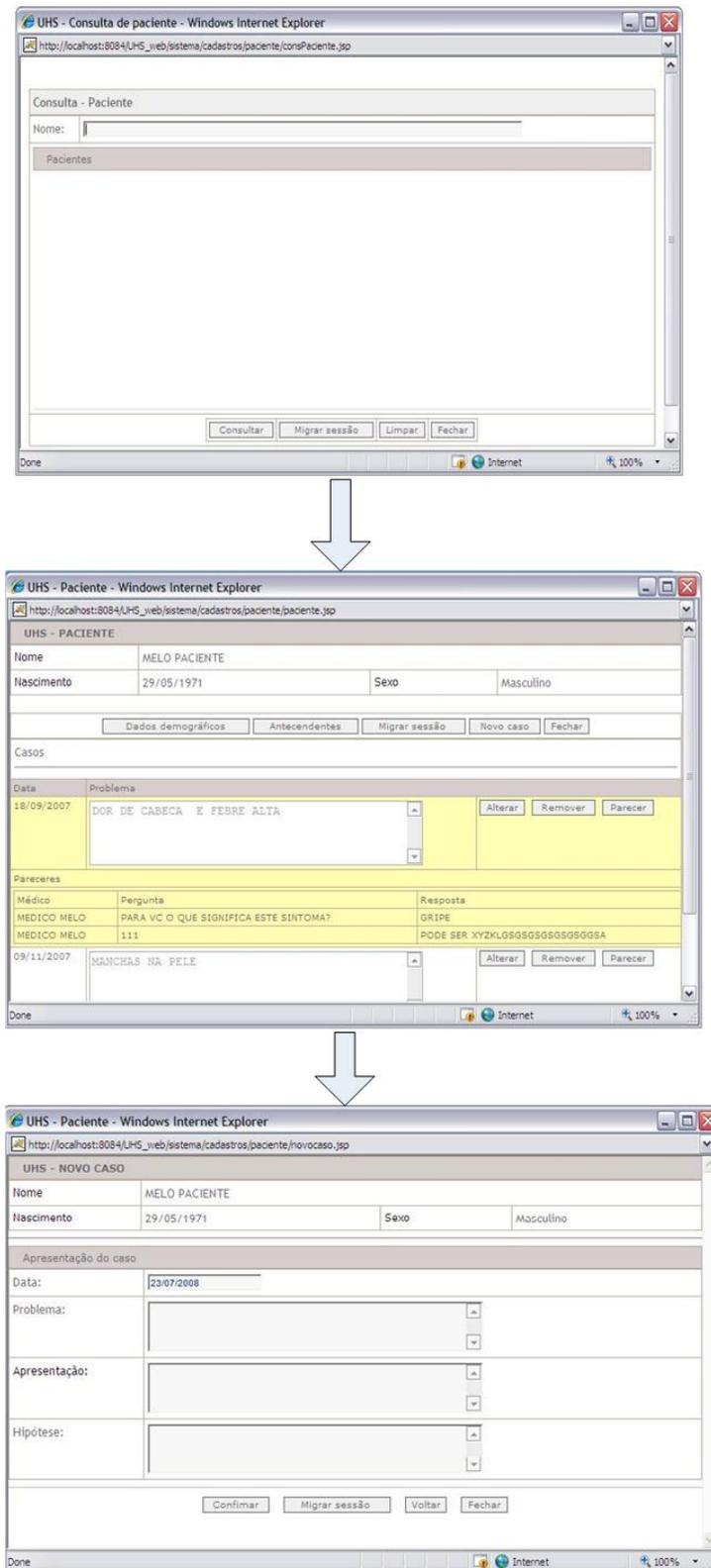


Figura 5.12 Telas para Consultar um Paciente Existente e seus Casos

A remoção de um caso também só estará habilitada para o usuário médico que o incluiu e ficará registrada no arquivo de log do ambiente, uma vez que os sistemas de prontuários eletrônicos não permitem a exclusão definitiva de informações, para evitar que dados importantes sejam apagados de forma equivocada.

Observa-se que na tela onde são apresentados os dados do paciente, existem 5 botões na parte superior. O primeiro deles refere-se aos dados demográficos do paciente. O segundo botão, ao ser pressionado, chama a tela de antecedentes do paciente. O terceiro botão está presente em ambas as telas, bem como nas demais telas do sistema, pois refere-se a possibilidade de migrar sessão de forma desejada, que no escopo desta tese, refere-se a migração *user-driven*. Neste caso, o paciente poderá resgatar, em seguida, a sessão usando qualquer dispositivo. O quarto botão possibilita que um novo caso clínico seja anexado ao PEP daquele paciente. A tela que representa a inclusão do caso clínico é a terceira tela da Figura 5.12. No momento da inclusão do caso, o médico descreve o problema e a apresentação do mesmo com os respectivos sintomas, colocando uma hipótese que é levantada por ele para aquele caso, confirmando, em seguida, a inclusão. Nesta tela também pode ser observado o botão que possibilita a migração da sessão, bem como as opções de voltar à tela anterior e limpar. O quinto e último botão apenas fecha a tela.

O terceiro item do menu da tela principal denomina-se "Médico". Ao escolher esta opção o usuário poderá responder um parecer (segunda opinião) que lhe foi solicitado ou fazer busca por médicos associados ao ambiente. Para buscar um médico no ambiente, o usuário poderá solicitar uma consulta por especialidade, status ou localização. Por exemplo, ele poderá consultar apenas médicos que estejam no Hospital da Restauração, com o status disponível e na especialidade Cardiologia. Caso o usuário deseje responder a um parecer, ele escolherá a tal opção no menu "Médico" e visualizará uma tela com todas as solicitações de pareceres que lhe foram enviadas. Esta tela está representada pela primeira tela da Figura 5.13. Quando ele pressiona o botão "Responder Parecer" a segunda tela da Figura 5.13 será apresentada para ele, possibilitando-o analisar o caso e responder ao parecer.

A última opção do menu da tela principal denomina-se "Tabelas Auxiliares". Através dessa opção, o médico poderá consultar os estabelecimentos credenciados ao ambiente UHS, como por exemplo, unidades de saúde e hospitais.

5.4.2 Interface Cliente para Telefones Celulares

Telefones celulares, por exemplo, são dispositivos muito utilizados para acesso ao ambiente UHS, embora ofereça pouco conforto para a inclusão de informações em grande volume de-

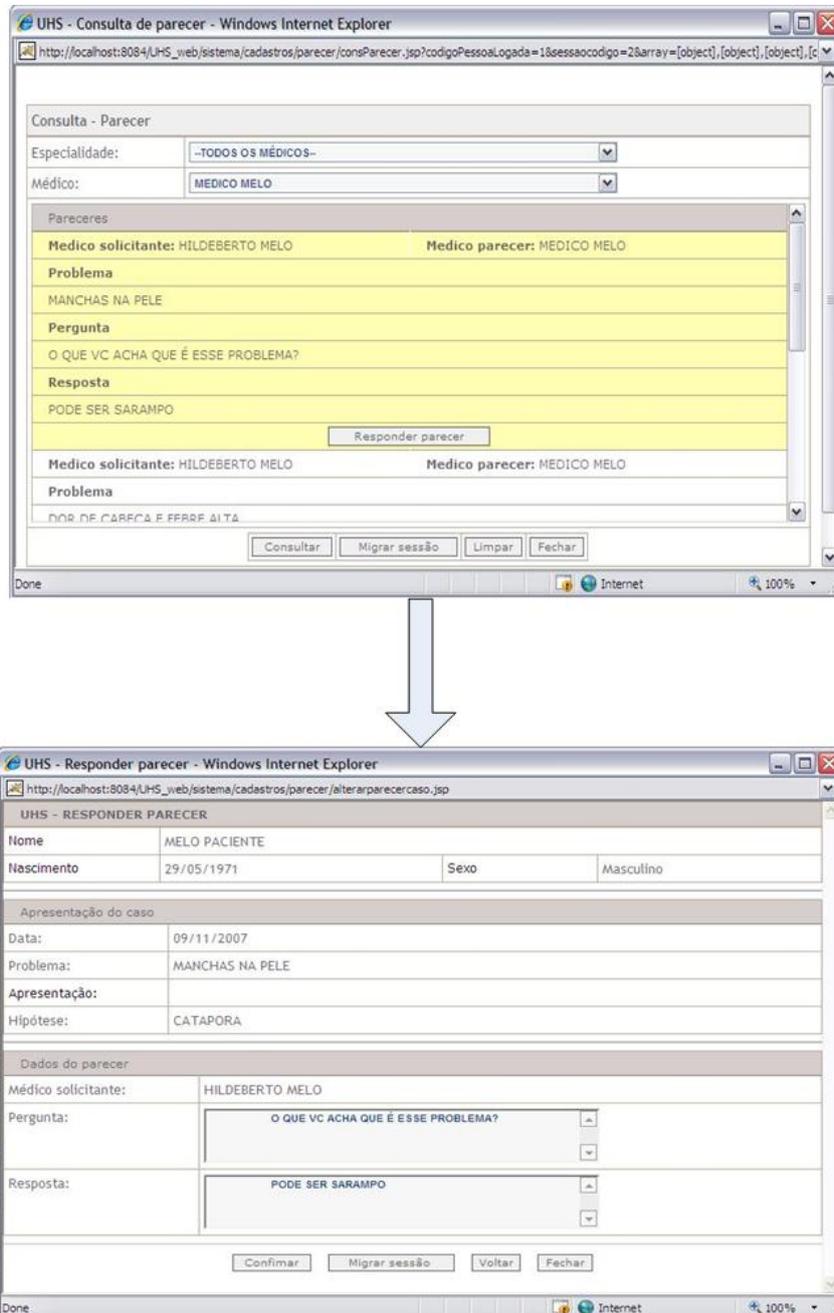


Figura 5.13 Telas para Consultar e Responder Pareceres

vido a sua interface e teclado limitados. Com base nisto, a interface *Micro Edition* parte do princípio que o médico não irá realizar inclusões de casos clínicos fazendo uso de um dispositivo de interface limitada, tendo em vista que tais dados são incluídos no momento em que o paciente é atendido em uma unidade de saúde (hospital ou posto de saúde). Sendo assim, tal funcionalidade não precisa ser disponibilizada para os médicos que realizam acesso através do celular, por exemplo.

Com base numa pesquisa realizada com médicos da UCMF (Unidade de Cardiologia e Medicina Fetal) do Real Hospital Português de Recife, foram escolhidas algumas das funcionalidades que seriam disponibilizadas através de uma interface em um dispositivo como telefone celular. Dessa forma, as atividades mais prováveis de serem realizadas, fazendo uso de um telefone celular, são atividades de consulta a dados do PEP, e solicitações e respostas a pareceres médicos em ações de segunda opinião. Baseando-se nisto, o ambiente UHS disponibiliza um módulo de acesso aos PEPs usando telefones celulares que fazem uso do serviço de adaptação de conteúdo para a realização de filtragens de informações a serem disponibilizados ao médico, quando o mesmo realiza um acesso através de um dispositivo celular.

O objetivo principal dessas filtragens de conteúdo é fazer com que apenas informações relevantes sejam apresentadas ao usuário para minimizar os problemas com limitações de telas e teclados de tais dispositivos. A tela principal do aplicativo-cliente com acesso através do celular pode ser observada na primeira ilustração da Figura 5.14. A primeira opção do menu principal refere-se aos dados do próprio médico conectado àquela sessão. A segunda opção levará o usuário ao acesso a funcionalidades do paciente, como por exemplo, acesso ao PEP de um paciente específico. Isto será possível através de uma tela de busca, onde o sistema irá localizar, no servidor de aplicação do PEP, o paciente solicitado. Após o retorno dessa requisição, o médico poderá confirmar sobre o nome do paciente retornado, usando um dos botões do celular e essa ação apresentará os dados daquele paciente na tela do celular, conforme apresentado na segunda ilustração da Figura 5.14. Para se ter acesso aos casos clínicos associados àquele paciente, pressiona-se o botão "Menu" da segunda ilustração e escolhe a opção "Casos". Também para a interface *Micro Edition* é possível verificar a existência do botão "Migrar" em todas as telas (exceto na primeira), possibilitando a migração da sessão para outro dispositivo. Na segunda tela, embora esta opção não apareça inicialmente, ele encontra-se disponível ao clicar o botão "Menu". Observa-se na terceira ilustração (Figura 5.14) que os casos clínicos associados àquele paciente serão mostrados numa espécie de menu contendo a data em que o caso clínico foi iniciado.

Para acessar um caso clínico específico, o médico escolherá o caso desejado e uma nova tela lhe será apresentada. Esta tela pode ser visualizada na primeira ilustração da Figura 5.15.



Figura 5.14 Tela Principal de acesso via Celulares

Ao pressionar o botão "Menu", o usuário poderá escolher entre migrar a sessão ou consultar os pareceres associados àquele caso específico. Quando ele opta pela migração, a sessão fica no serviço de gerenciamento de sessões até ser resgatada em outro dispositivo. Quando a opção é pela exibição dos pareceres vinculados ao caso, a tela apresentada na segunda ilustração da Figura 5.15 aparecerá no celular do médico. Ele poderá, ainda, responder a um parecer que lhe foi solicitado através da escolha do botão "Menu" e da opção "Responder Parecer". Para esta última escolha, será apresentada a terceira ilustração apresentada na Figura 5.15.



Figura 5.15 Tela de Casos e Pareceres via Celulares

5.5 Considerações Finais

O presente capítulo apresentou a implementação dos serviços do *UbiDoctor* e de um protótipo para um ambiente de PEP, denominado UHSys, que utiliza as bibliotecas do *middleware UbiDoctor* em sua infra-estrutura.

A prototipação dos componentes da arquitetura teve como objetivo verificar a viabilidade da mesma e permitir a realização de experimentos e, posteriormente, a sua avaliação.

A arquitetura é composta por 3 serviços de *middleware*, o servidor de aplicação e as aplicações clientes para as interfaces celular e Web. É importante ressaltar que as bibliotecas dos serviços do *middleware*, são desacopladas da aplicação, podendo então ser re-utilizadas em cenários que apresentem requisitos similares aos do UHS.

A criação de um protótipo para o ambiente UHS, envolvendo tanto uma aplicação de PEP, quanto a implementação dos serviços especificados para o *middleware UbiDoctor*, viabiliza a realização de testes de desempenho e ainda possibilita a obtenção das impressões dos usuários, com relação ao ambiente proposto. Os resultados obtidos com os testes de desempenho e as impressões dos usuários serão apresentadas no Capítulo 6 a seguir.

CAPÍTULO 6

Avaliação

Este capítulo apresenta os experimentos realizados com a arquitetura de serviços de *middleware*, o *UbiDoctor*, considerando o cenário UHS dando suporte à natureza nômade e fragmentada do trabalho médico. A primeira parte do capítulo apresenta os experimentos realizados para avaliar o desempenho do *UbiDoctor*, considerando o número de sessões gerenciadas e o tempo de migração das sessões, com e sem a necessidade de adaptação de conteúdo. A segunda parte refere-se às impressões obtidas por médicos, após utilizarem o ambiente UHS, através de uma pesquisa qualitativa.

6.1 Avaliação de Desempenho

Esta seção descreverá os testes de desempenho realizados com o propósito de medir os tempos de migração de sessões, em função do número de sessões simultâneas, controlados pelo serviço de gerenciamento de sessões do *UbiDoctor*.

Parte-se da hipótese que se o tempo de migração não for pequeno suficiente, a migração de sessões não motivará o médico a aumentar sua produtividade. Para isto, foram analisadas situações em que se fez necessária a realização de adaptação de conteúdo, devido a migração ter se dado para um dispositivo com tela menor, e também situações na qual o dispositivo alvo da migração possuía o mesmo tamanho de tela do dispositivo de origem da sessão (não sendo necessária a adaptação).

Para a realização dos testes, foi utilizado um computador com a seguinte configuração: processador Pentium 4 com 3 GHz, 512 MB de memória RAM e sistema operacional Windows XP Professional. Foi utilizado ainda um *notebook* Compaq Presario M2010US, com processador Intel celeron com 1.3 GHz, 512 MB de memória RAM e sistema operacional Windows XP Professional.

A distribuição dos componentes de software para a realização dos testes foi a seguinte:

- A Máquina 1 contém o Servidor de Aplicação UHS web, o Servidor Web Apache Tom

Cat 5, o Sistema de Gerenciamento de Banco de Dados PostGres 8 e os serviços do *middleware UbiDoctor* em execução.

- A Máquina 2 é um cliente que executa chamada ao ambiente UHS usando o Navegador Windows Explorer 7. Para simular a interface celular foi utilizado o emulador Sun JavaTM Wireless Toolkit for CLDC Versão 2.5.1, já que não havia disponibilidade de um celular com conexão Wi-Fi para a realização dos testes. A conexão via rede das operadoras de telefonia móvel também seria inviável, pois envolveria um alto custo operacional.

Nos experimentos realizados foram analisadas três situações principais. No primeiro experimento, foi medido quanto o tempo de migração de sessões aumenta em função do número de sessões simultâneas gerenciadas pelo *UbiDoctor*, ou seja, quanto o tempo de migração percebido pelos clientes aumenta em função do número de clientes/sessões concorrentes, considerando que os clientes resgatam as sessões em um dispositivo com configuração similar ao que estavam anteriormente.

No segundo experimento, as mesmas medições foram realizadas, porém considerando que os clientes resgatam as sessões em um dispositivo com tela maior que a que ele executara anteriormente. Por exemplo, o médico chega no hospital conectado a uma sessão, via celular, e deseja continuar trabalhando em um *desktop*, por julgar ser mais confortável de se trabalhar.

O terceiro experimento apresenta uma sessão que foi iniciada em um dispositivo de tela grande como um *desktop* e, em seguida, a mesma será migrada para um com tela pequena, como um telefone celular. Neste caso, a rotina de adaptação de conteúdo deverá ser executada, uma vez que as informações precisam ser adaptadas para serem apresentadas na tela do celular.

Vale a pena ressaltar que todos os testes realizados envolveram a mobilidade local. Embora a arquitetura proposta dê suporte a mobilidade remota, o tempo de migração neste caso não pôde ser medido, pois o tempo de deslocamento no ambiente remoto (de um hospital para o outro, por exemplo) é imprevisível.

Para cada um dos experimentos foram realizados testes onde o número de sessões simultâneas controladas pelo serviço de gerenciamento de sessões do *UbiDoctor* variou de 10 a 1000. Foram obtidas medições referentes ao tempo de migração de sessão no ambiente UHS, ou seja, o tempo gasto para que uma sessão possa estar pronta esperando para ser utilizada em um outro dispositivo. Este tempo de migração envolve o tempo de processamento no *middleware UbiDoctor*, tempo de transporte de dados pela rede, tempo de acesso ao banco de dados, tempo gasto pelo processamento nos servidores web e de aplicação. Observou-se ainda que o tempo de migração também varia, a depender da tela em que o usuário se encontra, no momento da migração da sessão. Caso a tela possua um maior número de campos, este tempo despendido

no processo de migração pode ser maior. Nos testes foram consideradas duas telas: (i) Tela de Consulta aos dados de um Paciente e (ii) Tela de Resposta a um Parecer (segunda opinião). A segunda tela apresenta um maior número de componentes em sua interface.

Para cada uma das situações apresentadas foram realizados três testes estatísticos utilizando o software SPSS (*Statistical Package for the Social Sciences*). O primeiro deles, refere-se a parte descritiva, onde os valores de Média, Mediana e Desvio Padrão (Triola 1999) (Meyer 1983) para o tempo de migração foram calculados e os histogramas foram construídos pelo software.

Em seguida foi realizado o teste de independência de **Qui-quadrado** (Stevenson 2001). O objetivo do teste de independência é verificar se há dependência entre duas variáveis medidas nas mesmas unidades experimentais. Um exemplo simples: deseja-se verificar se existe dependência entre a renda e o número de filhos em famílias de uma cidade. O teste de **Qui-quadrado** trabalha com duas hipóteses: a hipótese nula e a hipótese alternativa. A primeira trabalha com a inexistência da associação entre as variáveis de classificação, enquanto que a hipótese alternativa sugere que existe associação entre as variáveis de classificação. Quando a hipótese nula é aceita, sugere que as duas variáveis comparadas são independentes (Stevenson 2001). Para o caso específico desta tese, deseja-se saber se há independência entre as variáveis tempo de migração e número de sessões simultâneas.

O terceiro teste estatístico realizado foi o **Teste t** (Freund e Simon 2000). Segundo Freund e Simon (Freund e Simon 2000), este método é utilizado para testar se a diferença observada entre duas médias amostrais pode ser atribuída ao acaso ou se é estatisticamente significativa. Nesta tese, o **Teste t** foi utilizado para saber se a amostra utilizada se comporta similarmente à população.

E por fim, foi realizada a **análise de regressão linear** (Triola 1999) (Jain 1991) para verificar o aumento médio no tempo de migração para cada nova sessão gerenciada pelo *UbiDoctor*. Para a **análise de regressão linear** admite-se que o número de sessões simultâneas seja constante e não valores de variáveis aleatórias, e que para cada valor do número de sessões simultâneas, a variável a ser predita (tempo de migração) tenha uma distribuição cuja média pode ser representada por uma reta (Freund e Simon 2000). As subseções a seguir apresentam os três experimentos aos quais o *UbiDoctor* foi submetido e os resultados obtidos.

6.1.1 *Desktop para Desktop*

Este primeiro experimento considerou que o médico migrou, durante uma sessão de trabalho, de um *desktop*(ou *notebook*) para outro *desktop*. Essas situações acontecem, por exemplo,

quando o médico sai do desktop de uma sala para o da outra, dentro do hospital, ou quando ele migra a sessão do seu *notebook* (a bateria estava descarregando) para um *desktop*.

Os histogramas referentes a este primeiro experimento gerados pelo software SPSS, considerando a tela (i) e (ii) são ilustrados no Apêndice A. Cada histograma representa a frequência de ocorrência do tempo de migração para um número determinado de sessões (10, 100, 200, 400, 600 e 1000). Fixando o número de sessões simultâneas, cada experimento foi executado 60 vezes para se obter essa média.

A Figura 6.1 apresenta estes dados na forma de gráfico com a evolução do tempo médio de migração, variando-se o número de sessões simultâneas, no serviço de gerenciamento de sessões. A curva inferior representa os tempos para a tela de consulta (i) e a curva superior refere-se aos tempos de migração para a tela de resposta a um parecer (ii).

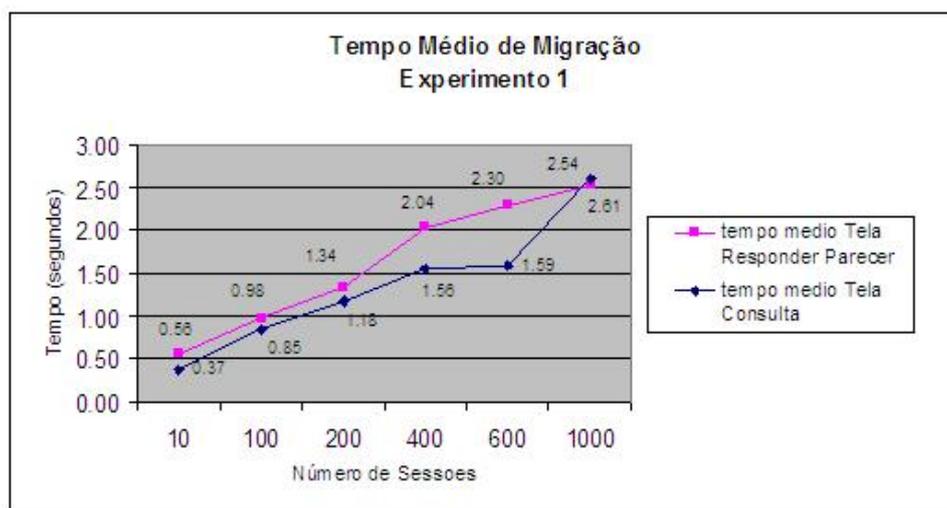


Figura 6.1 Tempo de migração versus Número de sessões

Para o número de sessões experimentadas, o tempo médio de migração para a tela (ii) é, em geral, superior para a tela (i). Observa-se, por exemplo, que o tempo médio para 10 sessões simultâneas é em torno de 0,56 segundos e para 1000 é 2,54 segundos, quando a migração foi solicitada, partindo-se da tela (ii). Para a tela (i), este valor é de 0,37 para 10 sessões e 2,61 para 1000 sessões simultâneas.

Os resultados mostram que não há independência entre as grandezas tempo de migração e número de sessões, e isto pode ser verificado a partir do segundo teste estatístico realizado, o teste de independência de **Qui-quadrado**. Observa-se nas Tabelas 6.1 e 6.2 que a significância do coeficiente **Person Qui-quadrado** é igual a zero para as duas telas utilizadas, rejeitando-se, portanto, a hipótese nula de independência entre as grandezas tempo de migração e número de

sessões simultâneas. Dessa forma, o teste confirma a hipótese de que há relação entre o tempo de migração e o número de sessões concorrentes, ou seja, quanto maior o número de sessões, maior o tempo de migração.

Tabela 6.1 Teste de Qui-Quadrado para o Experimento 1 - Tela Consulta Dados

	valor	Significância
Person Qui-quadrado	444.246	0
Número de Observações	360	

Tabela 6.2 Teste de Qui-Quadrado para o Experimento 1 - Tela Responder Parecer

	valor	Significância
Person Qui-quadrado	610.581	0
Número de Observações	360	

A terceira análise estatística realizada com o SPSS foi o **Teste t**. O **Test t** trata-se de um teste de diferença entre médias e atesta se as médias de dois grupos são estatisticamente diferentes uma da outra. As matrizes ilustradas nas Tabelas 6.3 e 6.4 apresentam os valores para a significância encontrada para o teste das diferenças entre médias. Compara-se as médias entre os 6 testes realizados (10, 100, 200, 400, 600 e 1000 sessões). Os valores da diferença entre as duas médias, juntamente com o valor da significância (entre parênteses) para as telas (i) e (ii), são colocados nas células equivalentes da matriz. Se a significância for menor que 0,05 (5 por cento), a probabilidade de se cometer um erro do tipo II, ou seja, aceitar a hipótese de trabalho ¹ quando ela não é verdadeira é menor que 5%. Em outras palavras, a significância, neste caso, representa a probabilidade de que a amostra utilizada se comporte diferentemente da população ². O valor 0,05 foi utilizado como referência para estes casos.

Apenas para os testes que comparam 400 X 600 sessões, a significância excede 5 por cento para a tela (i), indicando que a probabilidade de se ter escolhido uma amostra com comportamento diferente da população é superior ao desejado. Para a tela (ii), isto acontece para a comparação ente 400 X 600 e 600 X 1000. Todas as outras comparações entre médias, possuem significância inferior a 5 por cento, indicando que há grande probabilidade da amostra se comportar de maneira similar à população.

Para concluir a análise estatística, foi realizada uma **regressão linear por MQO** (mínimos quadrados ordinários) (Jain 1991). Através desta análise é possível identificar o aumento

¹Para esta tese, a hipótese de trabalho é a relação escolhida entre as variáveis, tempo de migração e número de sessões

²Em estatística, população refere-se a uma coleção de unidades individuais, que podem ser pessoas, animais, resultados experimentais, com uma ou mais características comuns, que se pretendem analisar.

Tabela 6.3 Teste das Diferenças entre Médias - Experimento 1

Teste T para a Tela Consulta Dados						
Teste t	10	100	200	400	600	1000
10		0.48 (0.0)	0.81 (0.0)	1.2 (0.0)	1.22 (0.0)	2.24 (0.0)
100			0.38 (0.03)	0.7 (0.0)	0.74 (0.0)	1.75 (0.0)
200				0.38 (0.03)	0.41 (0.01)	1.43 (0.0)
400					0.03 (0.85)	1.04 (0.0)
600						1.01 (0.0)
1000						

Tabela 6.4 Teste das Diferenças entre Médias - Experimento 1

Teste T para a Tela Responder Parecer						
Teste t	10	100	200	400	600	1000
10		0.42 (0.0)	0.78 (0.0)	1.48 (0.0)	1.74 (0.0)	1.98 (0.0)
100			0.36 (0.03)	1.06 (0.0)	1.32 (0.0)	1.56 (0.0)
200				0.70 (0.0)	0.96 (0.0)	1.2 (0.0)
400					0.26 (0.24)	0.5 (0.0)
600						0.24 (0.23)
1000						

médio no tempo de migração para cada nova sessão criada no serviço de gerenciamento de sessões. Na **regressão linear por MQO** obtém-se os coeficientes a_0 e a_1 da função da linha de tendência do tempo de migração. A reta que representa esta função possui a seguinte equação: $a_0 + a_1 * \text{número de sessões simultâneas}$.

Pode-se identificar que a cada nova sessão aberta há uma tendência de aumento do tempo médio de migração de 2 milissegundos tanto para a tela (i) quanto para a tela (ii). Este valor foi obtido através do uso do software SPSS e pode ser observado nas Tabelas 6.5 e 6.6. O valor encontrado de 2 milissegundos pode ser observado na coluna identificada por B . Este coeficiente representa o a_1 na equação da reta, enquanto que o a_0 refere-se ao valor apresentado na intersecção da primeira linha com a primeira coluna das respectivas tabelas.

Tabela 6.5 Regressão Linear para o Tempo de Migração - Experimento 1

Tela 1 - Consulta dados do Paciente				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.587	0.078	7.518	0.0
sessões	0.002	0.0	13.170	0.0

De acordo com os experimentos realizados, o tempo de migração total observado é pequeno, de modo a representar um baixo impacto no tempo observado entre duas atividades executadas

Tabela 6.6 Regressão Linear para o Tempo de Migração - Experimento 1

Tela 2 - Responder Parecer				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.864	0.079	10.981	0.0
sessões	0.002	0.0	12.870	0.0

sucessivamente. O valor de tempo médio considerado para a troca de atividades é 51 segundos, conforme mencionado no Capítulo 1, na página 2. Os valores para o tempo médio de migração encontrados ficam abaixo de 3 segundos, sendo pequenos o suficiente para não onerar o tempo decorrente entre duas atividades sucessivas (cerca de 6% do tempo médio de troca de atividades, considerando-se 51s). Caso se deseje descobrir qual o tempo médio de migração para um número qualquer de sessões (inclusive superior a 1000), basta aplicar a fórmula $a0 + a1 * \text{número de sessões simultâneas}$.

6.1.2 Celular para *Desktop*

O segundo experimento considerou que o médico inicialmente usou um celular para acessar o ambiente e suspendeu sua sessão e, em seguida, usou um *desktop* para resgatá-la. Essas situações acontecem, por exemplo, quando o médico chega em sua casa conectado a uma sessão e migra a sessão para o seu *desktop*, ou por falta de bateria, ou por julgar ser mais confortável.

Os histogramas referentes ao segundo experimento também estão ilustrados no Apêndice A. Como no primeiro experimento, cada histograma representa a frequência de ocorrência do tempo de migração para um número determinado de sessões (10, 100, 200, 400, 600 e 1000). A evolução do tempo de migração variando-se o número de sessões simultâneas, no serviço de gerenciamento de sessões, está representada no gráfico da Figura 6.2. A curva inferior representa os tempos para a tela de consulta (i) e a curva superior refere-se aos tempos de migração para a tela de resposta a um parecer (ii). Observa-se que os valores médios de tempo para a tela (i) são ligeiramente inferiores aos da tela (ii) para todos os números de sessões simultâneas testadas.

Os resultados também confirmam que não há independência entre as grandezas tempo de migração e número de sessões, e isto pode ser verificado a partir do segundo teste estatístico realizado, teste de **Qui-quadrado**. A significância do coeficiente *Person Qui-quadrado*, conforme observado nas Tabelas 6.7 e 6.8, é igual a zero para as duas telas utilizadas, rejeitando-se, portanto, a hipótese nula de independência entre as grandezas tempo de migração e número de sessões simultâneas.

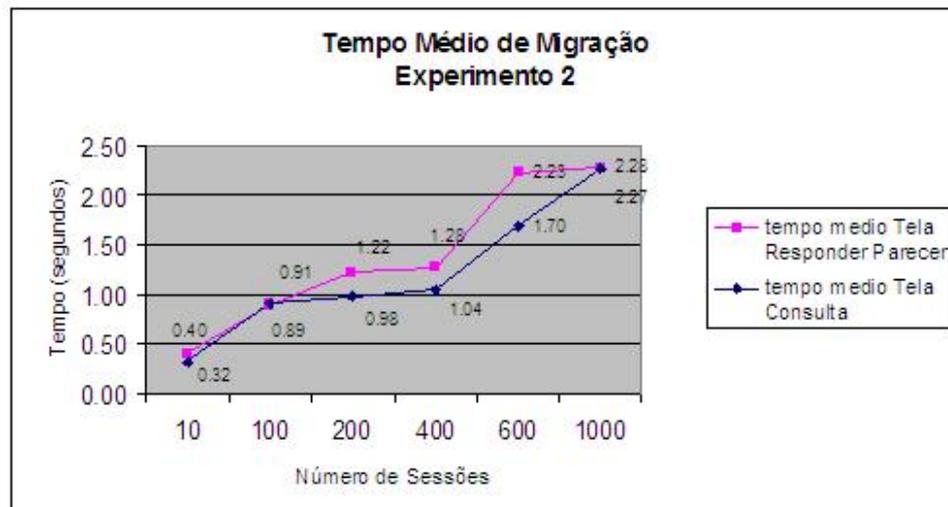


Figura 6.2 Tempo de migração versus Número de sessões

Tabela 6.7 Teste de Qui-Quadrado para o Experimento 2 - Tela Consulta Dados

	valor	Significância
Person Qui-quadrado	666.531	0
Número de Observações	360	

Tabela 6.8 Teste de Qui-Quadrado para o Experimento 2 - Tela Responder Parecer

	valor	Significância
Person Qui-quadrado	676.193	0
Número de Observações	360	

A terceira análise estatística foi o **Teste t** da diferença entre médias e atesta se as médias de dois grupos são estatisticamente diferentes uma da outra. As matrizes ilustradas nas Tabelas 6.9 e 6.10 apresentam os valores para a significância encontrada para o teste das diferenças entre médias. O valor da significância (entre parênteses) para as telas (i) e (ii) são colocados nas células equivalentes na matriz ilustrada para comparação de cada um dos testes variando-se de 10 a 1000 sessões, como no primeiro experimento. Se a significância for menor que 0,05, a probabilidade de se cometer um erro do tipo II, ou seja, aceitar a hipótese de trabalho quando ela não é verdadeira é menor que 5 por cento.

Para os testes que comparam 100 X 200, 100 X 400 e 200 X 400 sessões, a significância excede 5% para a tela (i), indicando que existe grande probabilidade de se ter escolhido uma amostra com comportamento diferente da população. Para a tela (ii), isto acontece para a comparação entre 200 X 400, e 600 X 1000. As demais comparações entre médias, possuem significância inferior a 5%, indicando que há grande probabilidade da amostra se comportar de maneira similar à população.

Tabela 6.9 Teste das Diferenças entre Médias - Experimento 2

Teste T para a Tela Consulta Dados						
Test t	10	100	200	400	600	1000
10		0.59 (0.0)	0.66 (0.0)	0.72 (0.0)	1.37 (0.0)	1.95 (0.0)
100			0.07 (0.67)	0.14 (0.3)	0.79 (0.0)	1.36 (0.0)
200				0.07 (0.63)	0.72 (0.0)	1.3 (0.0)
400					0.65 (0.0)	1.23 (0.0)
600						0.57 (0.0)
1000						

Tabela 6.10 Teste das Diferenças entre Médias - Experimento 2

Teste T para a Tela Responder Parecer						
Test t	10	100	200	400	600	1000
10		0.49 (0.0)	0.83 (0.0)	0.88 (0.0)	1.83 (0.0)	1.88 (0.0)
100			0.34 (0.0)	0.39(0.0)	1.34 (0.0)	1.4 (0.0)
200				0.05 (0.60)	1.00 (0.0)	1.06 (0.0)
400					0.95 (0.0)	1.00 (0.0)
600						0.05 (0.86)
1000						

Para concluir a análise estatística, foi realizada uma **regressão linear por MQO** (mínimos quadrados ordinários) identificando o aumento médio no tempo de migração para cada nova sessão criada no serviço de gerenciamento de sessões. Os testes confirmam os resultados do

primeiro experimento, de modo que a cada nova sessão aberta há uma tendência de aumento do tempo médio de migração de 2 milissegundos tanto para a tela (i) quanto para a tela (ii). Obtém-se este valor pelo software SPSS e pode ser observado nas Tabelas 6.11 e 6.12. O valor encontrado de 2 milissegundos pode ser observado na coluna identificada por *Unstandardized Coefficients - B*. A equação da reta que representa o tempo médio de migração é $0,515 + 0,002 * \text{número de sessões simultâneas}$ para a tela de consulta e $0,663 + 0,002 * \text{número de sessões simultâneas}$ para a tela de resposta a um parecer.

Tabela 6.11 Regressão Linear para o Tempo de Migração - Experimento 2

Tela 1 - Consulta dados do Paciente				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.515	0.054	9.552	0.0
sessões	0.002	0.0	16.928	0.0

Tabela 6.12 Regressão Linear para o Tempo de Migração - Experimento 2

Tela 2 - Responder Parecer				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.663	0.084	7.87	0.0
sessões	0.002	0.0	11.350	0.0

Para este segundo experimento, assim como para o primeiro, observa-se que o tempo de migração total observado é tão pequeno que representa um baixo impacto no tempo observado entre duas atividades executadas sucessivamente. Neste caso também há uma possibilidade de incentivar o médico a aumentar sua produtividade durante a realização de suas atividades de rotina.

6.1.3 Desktop para Celular

O terceiro e último experimento utiliza como base uma situação no qual o médico está trabalhando em um *Desktop* ou *notebook* e recebeu um chamado de urgência, solicitando, então, a migração da sessão para o seu celular. Ele poderá concluir o seu trabalho enquanto se desloca para atender o chamado, por exemplo. Nestes casos, em geral, o dispositivo de destino possui tamanho de tela inferior ao de origem, devendo, pois, ser chamada a rotina de adaptação de conteúdo, antes dos dados serem mostrados na tela do celular.

Espera-se, em geral, obter tempos de migração superiores aos experimentos anteriores, uma vez que a rotina de adaptação de conteúdo será chamada para a realização da filtragem de campos, aplicando-se as estratégias de adaptação citadas no Capítulo 4.

Os histogramas referentes ao terceiro experimento também estão ilustrados no Apêndice A. Como no primeiro experimento, cada histograma representa a frequência de ocorrência do tempo de migração para um número determinado de sessões (10, 100, 200, 400, 600 e 1000). A evolução do tempo de migração, variando-se o número de sessões simultâneas no serviço de gerenciamento de sessões, está representada no gráfico da Figura 6.3. A curva inferior representa os tempos para a tela de consulta (i) e a curva superior refere-se aos tempos de migração para a tela de resposta a um parecer (ii).

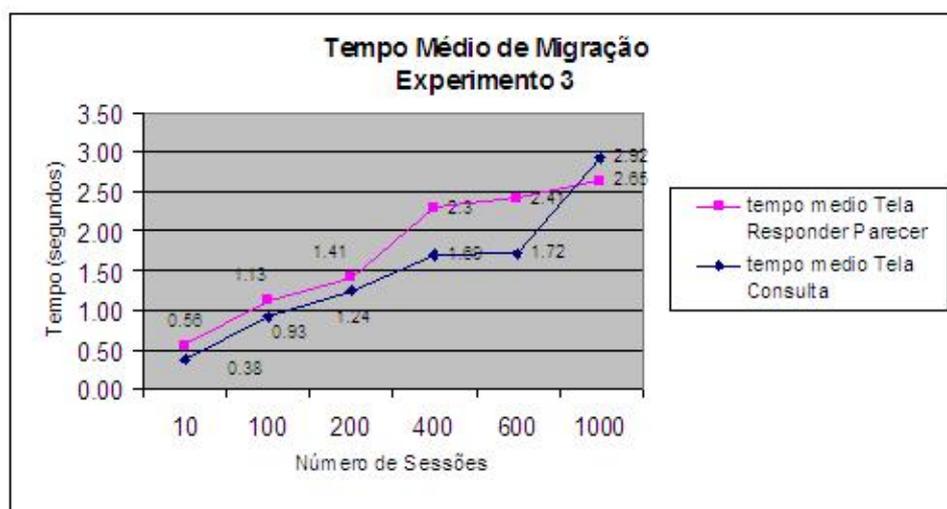


Figura 6.3 Tempo de migração versus Número de sessões

A significância do coeficiente *Person Qui-quadrado*, obtida com o teste estatístico de independência de **Qui-quadrado** (observada nas Tabelas 6.13 e 6.14) confirma que, também para este terceiro experimento, há dependência entre as grandezas tempo de migração e número de sessões. Isto leva à rejeição da hipótese nula de independência entre as grandezas tempo de migração e número de sessões simultâneas.

Tabela 6.13 Teste de Qui-Quadrado para o Experimento 3 - Tela Consulta Dados

	valor	Significância
Person Qui-quadrado	921.96	0
Número de Observações	360	

Quando realizado o **Teste t**, da diferença entre médias, pode-se observar se as médias de dois grupos são estatisticamente diferentes uma da outra. As matrizes ilustradas nas Tabelas 6.15 e 6.16 apresentam os valores para a significância encontrada para o teste das diferenças entre médias. Semelhante aos testes anteriores, se a significância for menor que 0,05, a proba-

Tabela 6.14 Teste de Qui-Quadrado para o Experimento 3 - Tela Responder Parecer

	valor	Significância
Person Qui-quadrado	610.581	0
Número de Observações	360	

bilidade de se cometer um erro do tipo II, ou seja, aceitar a hipótese de trabalho quando ela não é verdadeira, é menor que 5%.

Tabela 6.15 Teste das Diferenças entre Médias - Experimento 3

Teste T para a Tela Consulta Dados						
Test t	10	100	200	400	600	1000
10		0.55 (0.0)	0.87 (0.0)	1.32 (0.0)	1.34 (0.0)	2.54 (0.0)
100			0.31 (0.05)	0.76 (0.0)	0.79 (0.01)	1.99 (0.0)
200				0.45 (0.1)	0.48 (0.13)	1.68 (0.0)
400					0.03 (0.94)	1.23 (0.0)
600						1.20 (0.0)
1000						

Tabela 6.16 Teste das Diferenças entre Médias - Experimento 3

Teste T para a Tela Responder Parecer						
Test t	10	100	200	400	600	1000
10		0.56 (0.0)	0.84 (0.0)	1.73 (0.0)	1.85 (0.0)	2.09 (0.0)
100			0.28(0.31)	0.39(0.0)	1.34 (0.0)	1.4 (0.0)
200				0.89 (0.0)	1.00 (0.0)	1.24 (0.0)
400					0.11 (0.73)	0.35 (0.27)
600						0.24 (0.50)
1000						

Para os testes que comparam 200 X 400, 200 X 600 e 600 X 400 sessões, a significância excede 5 por cento para a tela (i), indicando que existe probabilidade de se ter escolhido uma amostra com comportamento diferente da população. Para a tela (ii), isto acontece para a comparação entre 100 X 200, 400 X 600, 400 X 1000 e 600 X 1000. As demais comparações entre médias, possuem significância inferior a 5%, indicando que há grande probabilidade da amostra se comportar de maneira similar a população.

A **regressão linear por MQO** (mínimos quadrados ordinários) foi o último procedimento realizado, com o objetivo de se identificar o aumento médio no tempo de migração, para cada nova sessão criada no serviço de gerenciamento de sessões.

Os testes confirmam os resultados anteriores, e para cada nova sessão aberta há uma tendência de aumento do tempo médio de migração de 2 milissegundos para ambas as telas. O valor encontrado de 2 milissegundos pode ser observado na coluna identificada por *B* das Tabelas 6.17 e 6.18 .

Tabela 6.17 Regressão Linear para o Tempo de Migração - Experimento 3

Tela 1 - Consulta dados do Paciente				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.6	0.122	4.922	0.0
sessões	0.002	0.0	9.58	0.0

Tabela 6.18 Regressão Linear para o Tempo de Migração - Experimento 3

Tela 2 - Responder Parecer				
Modelo	B	Erro Padrão	t	Significância
1 (constante)	0.955	0.128	7.456	0.0
sessões	0.002	0.0	8.149	0.0

Também para este experimento, pode-se extrapolar, aplicando-se os coeficientes na equação da reta e obtendo-se o tempo médio de migração para qualquer quantidade de sessões desejadas.

6.1.4 Conclusões Obtidas com os Experimentos

Os dados obtidos com os três experimentos são sintetizados nos gráficos da Figura 6.4. O gráfico superior representa a síntese dos três experimentos para a tela (i). As três curvas representam os experimentos realizados com a primeira tela. O gráfico inferior apresenta os resultados dos experimentos com a tela (ii) nos mesmos três experimentos. Os resultados obtidos levam a algumas conclusões importantes:

1. Em geral, os tempos decorridos no processo de migração da tela (ii) são superiores aos da tela (i), devido ao número de informações apresentadas na segunda tela ser maior que o da primeira.
2. O valor do tempo de migração médio aumenta, a medida que novas sessões vão sendo gerenciadas pelo serviço de gerenciamento de sessões. Isto pôde ser constatado a partir do teste estatístico de independência de **Qui-quadrado**, que comprova que não existe independência entre as grandezas tempo de migração e número de sessões, nos três experimentos realizados.

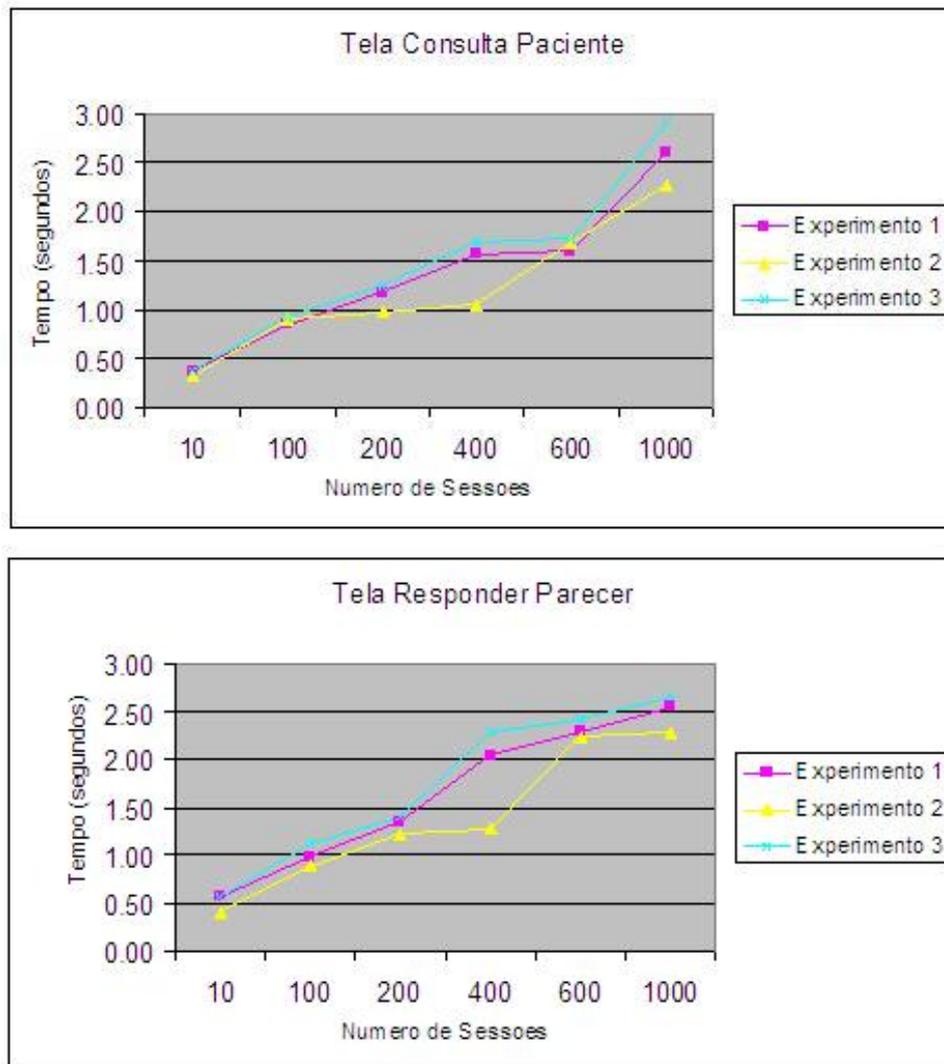


Figura 6.4 Síntese dos 3 Experimentos

3. O **Teste t** da diferença entre médias atestou, em boa parte dos casos, que as amostras coletadas representam, em geral, o comportamento estatístico da população. Isto comprova que a probabilidade de se ter escolhido uma amostra com comportamento estatístico diferente da população é inferior a 5 por cento. Vale ressaltar que algumas das diferenças entre médias superou os 5 por cento, e nestes casos específicos, a probabilidade que se tenha escolhido uma amostra com comportamento estatístico diferente da população, pode superar os 5 por cento.
4. Através da **regressão linear**, observa-se que para cada nova sessão gerenciada pelo *UbiDoctor* há um aumento médio de 2 milissegundos no tempo médio de migração de sessão.
5. Por fim, observa-se, em todos os experimentos realizados, que o tempo de migração total observado é pequeno, de modo a representar um baixo impacto no tempo observado entre duas atividades executadas sucessivamente. O valor de referência utilizado é 51 segundos (Tentori e Favela 2008). Os valores obtidos nos experimentos são inferiores a 3 segundos, o que acarreta um pequeno acréscimo sobre os 51 segundos (em torno de 6%) já demandados para a troca de atividades sucessivas. Supondo, por exemplo, que no momento em que um médico acessa dados num PEP, usando um desktop, ele é interrompido por um chamado de urgência no hospital. Ele poderá migrar sua sessão para o seu telefone celular, motivado pelo fato de que o tempo de migração é pequeno e concluí-la até chegar ao local de sua próxima atividade, possibilitando então, o aumento na produtividade do seu trabalho.

6.1.5 Limitações Observadas nos Experimentos

Durante a realização dos experimentos, os tempos de migração foram medidos considerando-se até 1000 sessões simultâneas gerenciadas pelo *UbiDoctor*. É provável que o tempo médio de migração tenha um impacto superior no tempo observado, entre duas atividades médicas consecutivas, a medida que o número de sessões simultâneas se torne muito superior a 1000.

Entretanto, imagina-se que essa limitação possa ser superada a partir do uso de uma arquitetura distribuída para o serviço de gerenciamento de sessões. De acordo com Coulouris e colegas (Coulouris G. ; Dollimore e Kindberg 2005), um serviço pode ser implementado por diversos processos servidores, em computadores separados. Os servidores podem hospedar o conjunto de objetos que o compõem de forma distribuída, ou então, pode-se manter cópias replicadas do serviço em diversos computadores. Isto ajudaria na obtenção de uma maior escalabilidade para o serviço de gerenciamento de sessões.

Este modelo de múltiplos servidores também poderia ser adotado para servidores web, contemplando o balanceamento de carga, para otimizar o atendimento às requisições do serviço web, providos pelo *Apache TomCat*.

6.2 Testes com Usuários

Toda solução tecnológica não pode perder de vista aqueles que vão validar a sua relevância, e no caso do presente trabalho, serão os usuários médicos. Nesta seção, serão descritos os resultados obtidos através de pesquisa qualitativa realizada com os usuários do *UbiDoctor*, através do protótipo desenvolvido para o cenário UHS.

A avaliação qualitativa é utilizada quando o fenômeno a ser estudado é complexo, de natureza social e não tende à quantificação (Neves 1996). De acordo com Denzin e Lincoln (Denzin e Lincoln 2005), a palavra qualitativa implica uma ênfase sobre as qualidades das entidades e sobre os processos e os significados que não são examinados ou medidos experimentalmente em termos de quantidade, intensidade, volume ou frequência. Os pesquisadores que aderem ao paradigma qualitativo estão focados na natureza socialmente construída da realidade. O paradigma qualitativo pode utilizar diferentes métodos, dentre os quais as entrevistas, questionários, a observação em campo, e a análise do discurso (Humphrey e Lee 2004). Para efeitos desta tese, será utilizado o método de entrevista, fazendo uso de um questionário com questões abertas e de múltipla escolha.

Para Queiroz (Queiroz 1988), a entrevista semi-estruturada é uma técnica de coleta de dados que supõe uma conversação continuada entre informante e pesquisador, e que deve ser dirigida por este de acordo com seus objetivos. Deste modo, da vida do informante só interessa aquilo que vem se inserir diretamente no domínio da pesquisa. Segundo Flick (Flick 2004), a entrevista semi-estruturada está sendo amplamente utilizada pela expectativa de que é mais provável que os pontos de vista dos sujeitos entrevistados sejam expressos em uma situação de entrevista com um planejamento relativamente aberto.

Nos experimentos realizados, são avaliados aspectos do uso do *UHSys*. Essa avaliação possui o intuito de identificar se o ambiente possibilita ao usuário médico a obtenção de ganhos de produtividade em seu trabalho.

6.2.1 Metodologia Adotada

Segundo Duarte (Duarte 2002), nas pesquisas de cunho qualitativo, a definição de critérios para se selecionar os sujeitos que vão compor o universo de investigação é algo primordial, pois interfere diretamente na qualidade das informações a partir das quais será possível construir a análise e chegar à compreensão mais ampla do problema delineado.

Para o presente trabalho, o questionário foi aplicado a um grupo de nove médicos cardiologistas que atuam em um hospital particular em Recife, Pernambuco, o Real Hospital Português. A aplicação do questionário foi realizada de forma individual, sem a interferência da pesquisadora, e se deu após a apresentação do protótipo ao médico. Em pesquisas qualitativas, utiliza-se normalmente pequenos grupos de entrevistados, pois não há um ganho adicional em se utilizar grupos maiores (Duarte 2002).

Com o objetivo de proteger a privacidade dos usuários utilizou-se um termo de compromisso que foi assinado por todos os participantes garantindo o seu anonimato e a confidencialidade de dados (nenhuma informação pessoal ou capaz de identificar o participante será divulgada pelos pesquisadores). Este termo pode ser encontrado no apêndice B desta tese. O questionário, por sua vez, foi dividido em onze perguntas que estão listadas no apêndice C e teve os seguintes objetivos:

1. Deseja-se, inicialmente, descobrir em que momentos são mais comuns as migrações de sessões. Com isto, objetiva-se observar se este processo de migração acontece dentro de um intervalo de tempo considerado satisfatório pelo médico.
2. Também se pretende descobrir se o conteúdo mostrado numa situação de migração, para um dispositivo alvo com tela menor que o que originou a sessão, é suficiente para a análise de um caso clínico através de um dispositivo móvel, como um telefone celular.
3. Por fim, deseja-se saber se de fato o uso dos serviços (de *middleware*) do *UbiDoctor*, dentro do protótipo, agregou a possibilidade de aumento de produtividade nas atividades executadas pelos médicos.

6.2.2 Perfil dos Participantes

Durante a realização das entrevistas foram identificadas três classes de indivíduos na amostra. O perfil dos usuários entrevistados foi bem diversificado contendo líderes de equipe com cerca de 30 anos de profissão até os residentes com cerca de 5 a 6 anos de medicina. Os nove usuários estão divididos, com relação aos anos de experiência profissional, nas seguintes classes:

- Quatro possuem menos de 10 anos;
- Dois possuem entre 10 e 20 anos;
- Três possuem mais de 20 anos;

Quando eles foram questionados sobre como consideravam o seu conhecimento em informática, seis pessoas classificaram-se como usuários com nível de conhecimento mediano. Apenas uma pessoa se colocou como um usuário com nível baixo de conhecimento e dois médicos classificaram-se como usuários avançados. Destes dois que se classificaram como usuários avançados, um deles possui 27 anos de profissão e o outro 4 anos. O médico com mais de 30 anos de profissão é o único que se considera um usuário com baixo conhecimento em informática. Em especial, observamos a presença de um médico com bastante experiência de profissão (27 anos), líder de equipe e com facilidade em lidar com a tecnologia. Provavelmente no futuro, médicos venham a ter um perfil similar ao deste usuário.

Dentre os usuários entrevistados, sete já fizeram uso de PEPs, pois enfatizaram que o Real Hospital Português já dispõe de sistema de PEPs e oferece treinamento aos seus colaboradores. Também são usuários de aplicações em dispositivos móveis, tais como PDAs e *smatphones*. As aplicações frequentemente utilizadas por estes usuários, em tais dispositivos, são Internet (e-mail, acesso a sites e downloads), jogos, planilhas e agendas. Todos os médicos consideraram que o uso de um sistema de informação que dê acesso a PEPs distribuídos nos diversos hospitais credenciados a uma rede facilitariam a sua vida, pois, poderiam acessar informações de pacientes, em locais remotos, para tomada de decisões, através do acesso a informações em tempo real. Eles consideram um ferramental importante já que desempenham uma trajetória nômade, trabalhando em vários hospitais. Dos 9 médicos, 3 já vivenciaram alguma situação na qual precisaram acessar PEPs em dispositivos móveis. Essas situações envolviam a necessidade de resgatar dados do paciente, consultar exames e emitir segunda opinião (solicitar ou responder parecer) sobre um caso clínico.

6.2.3 Resultados Obtidos

Os médicos foram questionados sobre a utilidade de um serviço que permitisse a migração de uma sessão em execução de um dispositivo para outro. Todos os entrevistados responderam positivamente a essa pergunta e selecionaram algumas das situações onde essa migração é mais útil. O questionário sugeriu algumas situações envolvendo migrações e o usuário poderia escolher. São elas: *dektop* para celular, celular para *dektop*, *desktop* para *desktop* e celular para celular. O usuário poderia assinalar todas, algumas ou nenhuma das situações.

Todos os usuários entrevistados marcaram a situação de migração entre dois *desktops* ou *notebooks*, na qual o médico *Precisa sair do consultório/hospital e gostaria de continuar o trabalho em casa, algumas horas depois.*

Oito usuários marcaram as opções abaixo, com relação à migração de um *desktop* para um celular ou entre dois celulares:

- *Precisa sair do consultório ou residência e gostaria de continuar o trabalho em seu celular;*
- *Continuará dentro do hospital, mas recebeu um chamado de urgência para atender um paciente e pode continuar o trabalho do seu celular, enquanto desloca-se ao encontro do paciente;*
- *A bateria do seu celular está acabando e precisa continuar o trabalho usando outro celular.*

Sete usuários assinalaram a opção *Deseja digitar muitas informações e prefere fazê-lo em um notebook ou computador de mesa*, em se tratando da migração de um celular para um *desktop*. Por fim, foi verificado que seis usuários selecionaram as três situações abaixo, no que diz respeito à migração celular para *desktop* e *desktop* para *desktop*:

- *Precisa continuar o trabalho e bateria do seu celular está acabando;*
- *Deseja continuar trabalhando em um dispositivo que permita melhor visualização do prontuário eletrônico de pacientes;*
- *Precisa sair do consultório, hospital ou residência e gostaria de continuar o trabalho em seu notebook no restaurante/refeitório.*

Os resultados que se referem a situações onde a migração de sessões torna-se necessária, comprovam a fragmentação da atividade e a mobilidade do médico e evidenciam que as mudanças de dispositivos, nas quais o dispositivo de origem é um *desktop* e o de destino é um celular, ou outro *desktop*, são as mais comuns. Também evidenciaram que a migração de um celular para outro seria útil.

Baseando-se nestas situações citadas, foi realizada uma simulação do ambiente operacional na qual foi possível o médico realizar a migração de um *desktop* para um celular e vice-versa. Em seguida, foi perguntado ao médico, como ele classificaria o processo de migração em relação à velocidade. Todos consideraram o processo rápido, a exceção de um usuário (11,11%) que o classificou como lento.

Em seguida, objetivou-se descobrir se o conteúdo mostrado na tela do celular foi suficiente para a análise de um caso clínico. O conteúdo a ser mostrado em alguns campos da tela do telefone celular, sofre adaptações de conteúdo, de acordo com as estratégias citadas no Capítulo 4. Sete médicos (77,78%) responderam que sim, que o conteúdo apesar de utilizar uma linguagem semelhante à linguagem de telegrama, oferece os subsídios necessários para uma boa análise do paciente. Um dos usuários, inclusive, fez o seguinte comentário ao justificar sua resposta: *"Fornece informações resumidas porém suficientes para dar segmento ao processo"*. Outro usuário justificou com o seguinte comentário: *"Pode dar uma visão global, já que contém campos suficientes para analisar um caso clínico, e se for necessário, terá que ir ao prontuário pela interface web"*.

Apenas dois médicos (22,22%) assinalaram a opção NÃO, quando questionados se o conteúdo mostrado dava informações suficientes para que ele analisasse o caso. Um destes usuários justificou que isto poderia ser sanado ao serem apresentados dados do exame físico (exemplo: olhos amarelados, pele com manchas) e, inclusive, sugeriu que fosse utilizada a linguagem de *chats* para a adaptação do conteúdo. Este usuário demonstrou, em seu perfil, estar familiarizado com essa linguagem e ser usuário dela em seu dia a dia. O outro usuário, insatisfeito com a adaptação, justificou que o espaço dos campos era insuficiente, mesmo com as abreviações para a sua análise do caso.

Para concluir, foi perguntado se os usuários acreditavam que o ambiente proposto possibilitaria aumentar a produtividade do seu trabalho. Apenas um usuário respondeu que não seria possível melhorar a sua produtividade com o ambiente proposto, embora quando perguntado se acredita que a migração de sessões, entre dispositivos, seja útil para a realização de seu trabalho, ele tenha respondido que sim. Este é um dos pontos importantes de uma pesquisa qualitativa, pois é possível revelar contradições e ir além dos números. Todos os demais (88,89%) responderam que a produtividade poderia ser aumentada, confirmando, pois, a hipótese deste trabalho.

6.2.4 Avaliação de Funcionalidades

Os usuários foram questionados com relação às funcionalidades que julgam importantes em PEPs, sobretudo quando utilizados a partir de um dispositivo móvel. Eles puderam classificá-las em essenciais, desejáveis e irrelevantes.

Dentre as funcionalidades essenciais, a maioria dos médicos selecionou as opções descritas na Tabela 6.19. Verifica-se o percentual de cada resposta na coluna denominada *Quantitativo* //

As funcionalidades consideradas desejáveis pela maioria dos médicos são observadas na

Tabela 6.19 Síntese de Funcionalidades Essenciais

Funcionalidade	Quantitativo
Inclusão de um novo paciente	66,67%
Consulta a dados e exames de um paciente	100%
Acréscimo de um novo caso clínico a um paciente	55,56%
Consulta a um caso clínico específico	88,89%
Acréscimo de informações a um dado caso clínico	77,78%

Tabela 6.20.

Tabela 6.20 Síntese de Funcionalidades Desejáveis

Funcionalidade	Quantitativo
Alterar seus dados pessoais	55,56%
Acrescentar informações aos dados demográficos do paciente	55,56%
Acrescentar informações aos antecedentes do paciente	66,67%
Solicitar segunda opinião	66,67%
Consultar pareceres de um caso clínico	66,67%
Responder os pareceres solicitados	55,56%
Visualizar o cadastro dos médicos no sistema	66,67%
Visualizar o cadastro de hospitais e unidades de saúde	88,89%

Além dessa classificação das funcionalidades, o questionário possuía um campo aberto onde o médico pôde colocar outras funcionalidades que julgasse essenciais ou desejáveis. Foram citadas as seguintes atividades: visualizar imagens de exames e dados de exame físico; obter dados de avaliação de equipe médica e de enfermagem para pacientes internos; solicitar exames; modificar prescrição médica e segurança.

Pode-se concluir com as respostas dos usuários, em relação às funcionalidades, que a interface do protótipo *UHSys* em dispositivos móveis, atende às necessidades dos médicos, uma vez que contempla a maioria das funcionalidades julgadas essenciais e desejáveis. Através do *UHSys*, o médico pode realizar as seguintes tarefas:

- Consultar e alterar seus dados pessoais;
- Consultar dados de um paciente cadastrado;
- Consultar um caso clínico específico;
- Acrescentar informações a um caso clínico existente;
- Solicitar segunda opinião;

- Consultar pareceres de um caso clínico;
- Responder pareceres solicitados.

6.2.5 Principais Conclusões dos Experimentos com Usuários

Nas entrevistas realizadas com uma equipe de médicos cardiologistas do Real Hospital Português foi possível observar que a depender do grau de conhecimento em informática do médico, ele vai estar mais familiarizado com o uso de dispositivos móveis. Entretanto, não necessariamente, os médicos mais experientes (pessoas de mais idade) são aqueles que menos utilizam os recursos tecnológicos. Isto pôde ser evidenciado a partir do questionário respondido por um médico, líder de equipe, com 27 anos de experiência, que apresentou características de um usuário avançado, em relação aos recursos tecnológicos disponíveis em sua profissão. Em geral, os médicos da equipe pesquisada, apresentam familiaridade com o uso de PEPs já que este recurso está sendo implementado em um dos hospitais em que eles atuam e por este motivo, eles estão recebendo treinamento neste tipo de ferramenta de trabalho.

De acordo com os objetivos 1, 2 e 3 discutidos na Seção 6.2.1, chega-se as seguintes conclusões:

- Conclusões obtidas com o objetivo 1 - momentos de migração:
 1. Todos evidenciaram a necessidade de mobilidade e interrupção de atividades devido a chamados de urgência. Isto pode ser ainda mais evidente, pois os entrevistados atuam em consultórios, na realização de exames de eco-cardiograma, em cirurgias e atendimento em UTIs (Unidades de Terapia Intensiva).
 2. Todos os entrevistados acreditam que a migração de sessões em execução, entre dispositivos, poderia facilitar a sua vida e selecionaram diversas situações onde isto seria interessante. Nenhuma das situações sugeridas no questionário foi descartada pelo grupo. Todos os médicos selecionaram pelo menos 3 situações sugeridas, nas quais o processo de migração seria interessante.
 3. Dentre os 9 médicos, 8 consideraram o processo de migração rápido.
- Conclusão obtida com o objetivo 2 - conteúdo suficiente: Os usuários apontaram que a filtragem de palavras, como uma alternativa para a adaptação do conteúdo a ser mostrado em um dispositivo de tela pequena, possibilita o entendimento das descrições contidas

em alguns campos do PEP. Apenas um dos usuários não acredita que essa seria uma alternativa que possibilitaria o entendimento dos casos clínicos. Todos os outros (88,89%) consideram interessante que algum tipo de estratégia seja utilizada para reduzir o número de palavras a serem apresentadas numa tela de um celular. Acredita-se que essa realidade é conhecida por eles, uma vez que, em sua maioria, são usuários de dispositivos móveis, conhecendo, pois, as limitações de telas e teclados de tais equipamentos. Uma sugestão importante foi apresentada por um dos médicos, com mais de 20 anos de experiência, quanto ao uso da linguagem de *chats* para a filtragem de conteúdo. Essa opção poderia ser incorporada ao serviço de adaptação de conteúdo e o médico poderia, inclusive, escolher que tipo de filtragem ele desejaria (supressão de palavras, a exemplo da linguagem de telegramas, ou o uso da linguagem de chats). É importante ressaltar que a interpretação sobre o conteúdo apresentado, no campo adaptado no celular, é de responsabilidade do médico, para que não seja obtida nenhuma conclusão ou diagnóstico errado sobre nenhum caso. Caso haja alguma dúvida por parte do médico, o mesmo deverá dirigir-se a um *desktop* para um diagnóstico mais completo.

- Conclusão obtida com o objetivo 3 - aumento de produtividade: Pôde-se concluir que os serviços propostos, sobretudo a possibilidade de migração em tempo hábil e com consistência de dados, quando testados em um protótipo, indicam que podem motivar o aumento na produtividade do profissional que apresenta características de mobilidade e fragmentação em suas atividades, sendo representado nesta tese pelo médico.

6.2.6 Limitações

A partir da demonstração do protótipo *UHSys* aos médicos foram identificadas algumas limitações, em termos de funcionalidades. Eles apontaram a necessidade de anexar imagens de exames laboratoriais, raio-x e vídeos de eco-cardiograma aos dados do paciente.

Um dos usuários salientou que também seria interessante verificar o comportamento do processo de migração de sessões quando a tela de trabalho do usuário, no momento da migração, for uma tela com uma imagem.

6.3 Considerações Finais

O presente capítulo apresentou duas formas de avaliação de resultados, obtidos com o uso de um protótipo, para um ambiente de PEP. O protótipo foi denominado *UHSys* e utiliza os

serviços do *middleware UbiDoctor* em sua infra-estrutura.

Com a análise quantitativa, observa-se que o tempo médio de migração ficou abaixo dos 3 segundos (para o caso de 1000 sessões) para os três casos observados. Os casos avaliados envolviam trocas de dispositivos dos seguintes tipos:

1. *Desktop* como dispositivo de origem e de destino;
2. Telefone celular como dispositivo de origem e *desktop* como dispositivo de destino;
3. *Desktop* como dispositivo de origem e telefone celular como dispositivo de destino;

Em cada uma das situações apresentadas foram experimentadas duas telas e os resultados obtidos foram submetidos a três testes estatísticos e a um processo de regressão linear. Concluiu-se que 3 segundos é um tempo pequeno se comparado a o tempo médio de duração de troca de atividades, conforme citado no Capítulo 1. Este tempo não desmotivará o médico a trocar de dispositivo para continuar numa mesma sessão. As medições realizadas foram referentes à mobilidade local, uma vez que o tempo de deslocamento no ambiente remoto é imprevisível. Embora o ambiente permita a mobilidade remota, o tempo de migração não pode ser considerado para os testes devido a característica de imprevisibilidade.

A segunda etapa da avaliação abrangeu uma pesquisa qualitativa realizada com potenciais usuários do protótipo e teve o objetivo de captar as impressões dos médicos em relação aos serviços e ao ambiente proposto.

Foi constatado, a partir das avaliações, que os serviços do *UbiDoctor*, quando utilizados em uma aplicação de PEP, possibilitam a migração de sessões em execução, entre dispositivos, em um intervalo de tempo pequeno, comparado aos 51 segundos em média, desprendidos entre duas atividades consecutivas, conforme estudo de observação realizado por Tentori e Favela (Tentori e Favela 2008). O tempo levado no processo de migração foi considerado aceitável pelos médicos questionados, de acordo com o estudo de impressões dos usuários, apresentado no presente capítulo. Também foram aceitáveis, os procedimentos de adaptação de conteúdo sugeridos para que o ambiente possibilite o uso de PEPs em dispositivos de tela pequena, como os telefones celulares. Essa pesquisa qualitativa apresenta-se como um indicativo de que os médicos são motivados a agirem mais rapidamente, possibilitando-os a obter ganhos de produtividade.

Conclusão

A computação ubíqua possui capacidade para atender grande parte das necessidades das aplicações de diversas áreas do mundo real. Um setor importante no qual as tecnologias ubíquas têm sido aplicadas é o domínio dos cuidados com a saúde. Como citado previamente, em ambientes de saúde, os usuários médicos estão constantemente em movimento, seja deslocando-se entre setores de um hospital ou no ambiente externo ao mesmo. O ambiente do escopo dessa proposta é voltado para auxiliar sobretudo médicos em sua rotina diária nos hospitais e consultórios, no que diz respeito ao uso de PEPs. A rotina deste profissional envolve nomadismo e fragmentação de atividades, conforme apontado nos capítulos iniciais desta tese.

Uma das formas de prestar suporte ao nomadismo dos médicos é permitir que ele continue executando sua atividade em um PEP, mesmo quando se desloca, possibilitando que ele migre a sua sessão de um dispositivo para outro, em tempo hábil e com garantias de consistência de informações.

De acordo com os desafios apontados anteriormente, as soluções para prover migração de sessões abordam apenas a mobilidade local, enquanto o presente trabalho aborda a mobilidade local e a mobilidade remota. Além disto, esta tese preocupa-se com o intervalo de tempo em que a migração da sessão acontece. Caso este tempo não seja pequeno o suficiente, ele poderá apresentar impacto sobre a produtividade do profissional, que já leva em média 51 segundos na mudança de uma atividade para outra (Tentori e Favela 2008).

Para permitir a migração de sessões entre dispositivos de características diferentes, faz-se necessário o uso de estratégias de adaptação de conteúdo, já que alguns dispositivos apresentam diferentes características, entre elas, tamanhos de tela.

Com o objetivo de superar os desafios apresentados, essa tese propôs uma arquitetura sensível a contexto que gerencia sessões com possibilidade de migração entre dispositivos, com características diferentes, em tempo hábil e com adaptação de conteúdo. O uso de tal arquitetura em um sistema de apoio a atividades de PEP irá possibilitar que a migração de sessão entre dispositivos aconteça de modo a causar um baixo impacto no tempo de troca de atividades e, conseqüentemente, motivem o médico a aumentar a sua produtividade perdendo menos tempo nesta troca.

Para projetar essa arquitetura, imaginou-se o cenário UHS descrito no Capítulo 1, onde médicos fazem acesso a PEPs em diversos hospitais de uma rede. O acesso a este ambiente é realizado de modo ubíquo (qualquer hora e qualquer lugar) e conta com os serviços do *middleware UbiDoctor* em sua arquitetura. Os serviços do *UbiDoctor* consistem de módulos de gerenciamento de sessão, gerenciamento de contexto e adaptação de conteúdo. Os serviços do *UbiDoctor* e o protótipo UHSys foram implementados usando a linguagem de programação java e utilizou *sockets* e invocação remota de métodos para implementar a comunicação entre os componentes distribuídos.

Foi proposta uma solução baseada em *middleware*, na qual seus serviços localizam-se na subcamada de serviços comuns, podendo ser re-usados em outros ambientes com características e requisitos similares. Dessa forma, outros aplicativos poderão fazer uso do *UbiDoctor*, reduzindo então a complexidade no desenvolvimento deste tipo de aplicação.

Para realizar a 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 pode-se observar qual era o impacto do tempo de migração de sessões considerando-se três experimentos. Para o primeiro experimento, considerou-se o uso de um desktop para suspender e outro para resgatar a sessão. No segundo experimento, considerou-se um celular 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 celular para resgatá-la. Para cada um destes experimentos, o tempo de migração foi observado para duas atividades avaliadas (Consultar paciente e Responder parecer). 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 e Favela 2008).

De acordo com os experimentos realizados, o tempo de migração total observado é tão pequeno, 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 *desktop* 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 poderá possibilitar um aumento na produtividade durante a realização de suas atividades de rotina.

Com a realização da pesquisa qualitativa, foi possível observar quais as impressões do usuário com o uso de uma ferramenta de PEPs que utiliza como base o conjunto de serviços disponibilizados pelo *UbiDoctor*. Essa pesquisa levou em consideração o grau de familiaridade dos usuários com dispositivos móveis e a experiência deles na profissão.

Foi possível verificar em que situações os médicos solicitariam a migração de aplicações e se eles julgavam que o processo aconteceu em um tempo satisfatório. Também foi constatado que para os usuários, este processo aconteceu dentro de um intervalo de tempo satisfatório, o que os levou a responder, quando questionados, que o ambiente possibilitaria a obtenção de ganhos de produtividade.

Tais conclusões reforçam a hipótese de que se o tempo de migração de sessão é pequeno o suficiente, ainda que seja necessário adaptar conteúdo, os médicos acreditam em possibilidades de ganho de produtividade.

7.1 Principais Contribuições

O presente trabalho apresentou uma arquitetura de *middleware* e serviços que permite a migração de sessões de aplicação (exemplo PEP) entre dispositivos diferentes, usando sensibilidade ao contexto para sugerir mudanças de dispositivos, adaptar conteúdo, possibilitando melhorias de produtividade do médico ou outro usuário que apresente características de nomadismo e fragmentação em suas atividades (ex professor).

Diante do que já foi proposto e construído, as principais contribuições desta tese são:

- Arquitetura de serviços *UbiDoctor* que possibilita:
 - acessar um ambiente de PEPs de forma ubíqua, permitindo iniciar uma sessão e realizar a sua suspensão e migração para outro dispositivo, com tempos de migração satisfatoriamente pequenos e, conseqüentemente, de baixo impacto na troca de atividade;
 - adaptar o conteúdo da sessão, favorecendo o uso de diversos dispositivos com diferentes configurações, baseando-se em informações de contexto;
 - gerenciar as sessões de PEPs de diferentes usuários, usando diferentes dispositivos e redes de acesso;
 - possibilitar a melhoria da produtividade de profissionais que tenham como característica a mobilidade e a fragmentação, através da redução do tempo de troca de atividades.
- Estratégias de adaptação de conteúdo baseadas em contexto;
- Esquema de caches que agrega valor no quesito confiabilidade do *middleware*;

- Processo de migração de sessões de baixo impacto possibilitando um aumento da produtividade do serviço médico;
- Comprovação da viabilidade dos serviços através da realização de análises qualitativa e quantitativa;

7.2 Trabalhos Futuros

Este trabalho poderá servir de base para novas pesquisas. Dentre elas estão:

- Ampliar o protótipo para contemplar informações multimídia (exames de ultrassom, vídeos em geral, etc) nos prontuários de pacientes e realizar experimentos para verificar o tempo de migração de sessões quando o processo inicia em uma sessão multimídia. Poderão ser realizadas melhorias na interface para ambos os dispositivos e incorporar funcionalidades que envolvam a manipulação de imagens e vídeos. Para tal, também poderão ser aprimorados os serviços do *UbiDoctor*, no sentido de otimizar o seu comportamento, quando trata-se de sessões que envolvam imagens e mídias contínuas.
- Implementar no protótipo o módulo que dá suporte a sessões com interação síncrona entre múltiplos usuários;
- Acrescentar ao serviço de adaptação, a estratégia de adaptação de conteúdo possibilitando o uso de linguagem de chats nos campos cujo conteúdo deve ser adaptado, conforme sugerido durante a pesquisa com usuários.
- Incorporar ao protótipo as funcionalidades e modificações sugeridas pelos médicos;
- Analisar a possibilidade de uso do *UbiDoctor* em outras aplicações sensíveis ao contexto com características similares.
- Incorporar padrões de terminologia semântica médica ao serviço de adaptação de conteúdo;
- Reprojetar o serviço de gerenciamento de sessões de modo a contemplar uma arquitetura distribuída, proporcionando assim, melhor robustez;
- Projetar mecanismos de segurança para permitir acesso aos PEPs fora do hospital;
- Projetar estratégia para facilitar a integração da informação dos PEPs.

7.3 Publicações do Trabalho Realizado

1. WebMedia 2007 - J. R. B. Diniz, and C. A. G. Ferraz. Gerenciamento de Sessão e Adaptação de Conteúdo em Medicina Ubíqua. In: WebMedia 2007 - XIII Simpósio Brasileiro de Sistemas Multimídia e Web, 2007, Gramado, RS. Proceedings WebMedia 2007.
2. ACM SAC 2008 - J. R. B. Diniz, C. A. G. Ferraz, and H. do N. Melo. An Architecture of Services for Session Management and Contents Adaptation in Ubiquitous Medical Environments. In: ACM Symposium on Applied Computing, 2008, Fortaleza. Proceedings 23rd ACM Symposium on Applied Computing, 2008. v. 2.
3. WebMedia 2008 - J. R. B. Diniz, F. A. M. Trinta, H. do N. Melo, L. M. Santos and C. A. G. Ferraz. Avaliação de um Serviço de Gerenciamento de Sessão para Ambientes de Medicina Ubíqua. In: WebMedia 2008 - IX Simpósio Brasileiro de Sistemas Multimídia e Web, 2008, Vila Velha ES, Proceedings WebMedia 2008.

Referências Bibliográficas

- [Abowd et al. 1999]ABOWD, G. D. et al. Towards a better understanding of context and context-awareness. In: *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag, 1999. p. 304–307. ISBN 3-540-66550-1.
- [Ahmed e Khan 2006]AHMED, M. S. S.; KHAN, A. J. Healthcare aide: Towards a virtual assistant for doctors using pervasive middleware. In: *PERCOMW '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, 2006. p. 490. ISBN 0-7695-2520-2.
- [Augustin 2004]AUGUSTIN, I. *Abstrações para uma Linguagem de Programação Visando Aplicações Móveis em um Ambiente de Pervasive Computing*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Instituto de Informática, 2004.
- [Bahl e Padmanabhan 2000]BAHL, P.; PADMANABHAN, V. N. Radar: An in-building rf-based user location and tracking system. In: *Proceedings of IEEE Infocom 2000*. [S.l.: s.n.], 2000.
- [Bardram 2005]BARDRAM, E. The trouble with login: on usability and computer security in ubiquitous computing. *Personal Ubiquitous Comput.*, Springer-Verlag, London, UK, v. 9, n. 6, p. 357–367, 2005. ISSN 1617-4909.
- [Bardram 2003]BARDRAM, J. E. Hospitals of the future: Ubiquitous computing support for medical work in hospitals. In: *Proceedings of the 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*. [S.l.: s.n.], 2003.
- [Bardram 2004]BARDRAM, J. E. Applications of context-aware computing in hospital work: examples and design principles. In: *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: ACM Press, 2004. p. 1574–1579. ISBN 1-58113-812-1.

- [Bardram e Christensen 2001]BARDRAM, J. E.; CHRISTENSEN, H. B. *Middleware for Pervasive Healthcare: A White Paper*. [S.l.], 2001.
- [Bardram e Christensen 2007]BARDRAM, J. E.; CHRISTENSEN, H. B. Pervasive computing support for hospitals: An overview of the activity-based computing project. *Personal Ubiquitous Comput.*, v. 6, n. 1, p. 44–51, 2007. ISSN 1536-1268.
- [Bardram J.; Kjær e C. 2003]BARDRAM J.; KJÆR, T. A. K.; C., N. Human-computer interaction with mobile devices and services. In: _____. [S.l.]: Springer Berlin / Heidelberg, 2003. cap. Supporting Local Mobility in Healthcare by Application Roaming Among Heterogeneous Devices, p. 161–176.
- [Bazire e Brézillon 2005]BAZIRE, M.; BRÉZILLON, P. Understanding context before using it. In: VERLAG, S. (Ed.). *CONTEXT-05 - 5th International and Interdisciplinary Conference*. [S.l.: s.n.], 2005. p. 29–40.
- [Bellotti e Bly 1996]BELLOTTI, V.; BLY, S. Walking away from the desktop computer: Distributed collaboration and mobility in a product design team. In: *Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work*. [S.l.]: ACM Press, 1996. p. 209–218.
- [Bluetooth.org 2004]BLUETOOTH.ORG. *The Official Bluetooth Membership Site*. 2004. [Http://www.bluetooth.org/spec/](http://www.bluetooth.org/spec/).
- [Brown P.; Bovey e Chen 1997]BROWN P.; BOVEY, J.; CHEN, X. Context-aware applications: From the laboratory to the marketplace. In: IEEE (Ed.). *IEEE Personal Communications*. [S.l.: s.n.], 1997. v. 4, n. 5, p. p. 58–64.
- [Brézillon]BRÉZILLON, P. *Modeling and Using Context: Past, Present and Future*. Disponível em: <citeseer.ist.psu.edu/549676.html>.
- [Buchholz e Schill 2003]BUCHHOLZ, S.; SCHILL, A. *Adaptation-aware web caching: Caching in the future pervasive web*. 2003. Disponível em: <[citepseer.ist.psu.edu/buchholz03adaptationaware.html](http://citeseer.ist.psu.edu/buchholz03adaptationaware.html)>.
- [Calv C. Z.; Pessoa e Filho 2005]CALV C. Z.; PESSOA, R. M.; FILHO, J. G. Um interpretador de contexto para plataformas de serviços context-aware. In: *Anais do XXV Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2005.
- [Cavalcanti et al. 2005]CAVALCANTI, D. et al. Issues in integrating cellular networks w lans, and manets: a futuristic heterogeneous wireless network. In: *IEEE Wireless Communications*. [S.l.: s.n.], 2005. v. 12, Issue: 3.

- [CFM 2002]CFM. *RESOLUÇÃO Conselho Federal de Medicina nº 1.638/2002*. 2002.
- [Chalmers 2002]CHALMERS, D. *Contextual Mediation to Support Ubiquitous Computing*. Tese (Doutorado) — Imperial College of Science, Technology and Medicine, Universidade de Londres, 2002.
- [Chen 2004]CHEN, H. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. Tese (Doutorado) — University of Maryland, 2004.
- [Coulouris G. ; Dollimore e Kindberg 2005]COULORIS G. ; DOLLIMORE, J.; KINDBERG, T. Distributed systems - concepts and design. In: _____. 4th. ed. [S.l.]: Addison Wesley, 2005. cap. 6, p. 657–719.
- [Coutaz 2005]COUTAZ, J. Context is key. *Communications of the ACM*, v. 48, n. 3, p. 49–53, março 2005.
- [Crowley et al. 2002]CROWLEY, J. L. et al. Perceptual components for context aware computing. In: *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*. London, UK: Springer-Verlag, 2002. p. 117–134. ISBN 3-540-44267-7.
- [Denzin e Lincoln 2005]DENZIN, N.; LINCOLN, Y. S. *The SAGE Handbook of Qualitative Research*. [S.l.]: Sage Publications, 2005.
- [Dey 2001]DEY, A. K. Understanding and using context. In: *Personal Ubiquitous Computing*. [S.l.]: Springer-Verlag London Ltd., 2001. v. 5, n. 1, p. 4–7.
- [Dix et al. 2000]DIX, A. et al. Exploiting space and location as a design framework for interactive mobile systems. *ACM Trans. Comput.-Hum. Interact.*, ACM Press, New York, NY, USA, v. 7, n. 3, p. 285–321, 2000. ISSN 1073-0516.
- [Dockhorn 2003]DOCKHORN, P. C. *"Towards a Services Platform for Context-Aware Applications"*. Dissertação (Mestrado) — University of Twente. Enschede, The Netherlands, 2003.
- [Duarte 2002]DUARTE, R. Pesquisa qualitativa: reflexões sobre o trabalho de campo. *CADERNOS de Pesquisa*, n. 115, Mar 2002.
- [Dustdar e Rosenberg 2004]DUSTDAR, M. B. S.; ROSENBERG, F. A survey on context-aware systems. In: *International Journal of Ad Hoc and Ubiquitous Computing*. [S.l.: s.n.], 2004.
- [Flick 2004]FLICK, U. *An Introduction to Qualitative Research*. [S.l.]: Bookman, 2004.

- [Fox 1989]FOX, C. A stop list for general text. *SIGIR Forum*, ACM, New York, NY, USA, v. 24, n. 1-2, p. 19–21, 1989. ISSN 0163-5840.
- [França e Ferraz 2007]FRANÇA, S. V. A.; FERRAZ, C. A. G. Signus: Arquitetura para suporte à ciência de contexto em sistemas de informações geográficas ubíquos. In: *Proceedings of WEBMEDIA 2007 - XIII BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB*. [S.l.: s.n.], 2007.
- [Freund e Simon 2000]FREUND, J. E.; SIMON, G. A. *Estatística Aplicada*. [S.l.]: Bookman, 2000.
- [Furuie et al. 2003]FURUIE, S. S. et al. Prontuário eletrônico de pacientes: integrando informações clínicas e imagens médicas. *Revista Brasileira de Engenharia Biomédica, SBEB - Sociedade Brasileira de Engenharia Biomédica*, v. 19, n. 3, p. 125–137, dec 2003.
- [Garlan D.; Steenkiste e Schmerk 2002]GARLAN D.; STEENKISTE, P.; SCHMERK, B. Project aura: toward distraction-free. In: *IEEE Pervasive Computing*. [S.l.: s.n.], 2002.
- [Goularte 2003]GOULARTE, R. *Personalização e adaptação de conteúdo baseadas em contexto para TV interativa*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP. USP - São Carlos, 2003.
- [Grimm e Bershad 2002]GRIMM, R.; BERSHAD, B. *Future directions: System Support for Pervasive Applications*. 2002. Disponível em: <citepseer.ist.psu.edu/grimm02future.html>.
- [Gu et al. 2004]GU, T. et al. A middleware for building context-aware mobile services. In: *Proceedings of IEEE Vehicular Technology Conference*. [S.l.: s.n.], 2004.
- [Handlye e Jacobson 1998]HANDLYE, M.; JACOBSON, V. *SDP: Session Description Protocol*. [S.l.], 1998.
- [Henricksen e Indulska 2005]HENRICKSEN, K.; INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing Journal*, 2005.
- [Hiltunen e Schlichting 2001]HILTUNEN, W.-K. C.; SCHLICHTING, R. Constructing adaptive software in distributed systems. In: *In 21st International Conference on Distributed Computing Systems*. [S.l.: s.n.], 2001.
- [Humphrey e Lee 2004]HUMPHREY, C.; LEE, B. *The Real Life Guide to Accounting Research a Behind-the-Scenes View of Using Qualitative Research Methods*. [S.l.]: Elsevier, 2004.

- [IEEE 2005]IEEE. *IEEE 802.11, The Working Group Setting the Standards for Wireless LANs*. maio 2005. [Http://grouper.ieee.org/groups/802/11/](http://grouper.ieee.org/groups/802/11/).
- [IEEE 2008]IEEE. *IEEE 802.15.6, IEEE 802.15 WPAN Task Group 6 Body Area Networks (BAN)*. novembro 2008. [Http://www.ieee802.org/15/pub/TG6.html](http://www.ieee802.org/15/pub/TG6.html).
- [Jain 1991]JAIN, R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. [S.l.]: Wiley, 1991.
- [Johnson 2005]JOHNSON, T. *Uma Arquitetura de Computação Pervasiva para Trabalho de Campo*. Tese (Doutorado) — Universidade Federal de Pernambuco, Março 2005.
- [Lee et al. 2002]LEE, J. Y. et al. Implementation of the session manager for a stateful server. In: *IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*. [S.l.: s.n.], 2002.
- [Lei e Nicolas 2001]LEI, Z.; NICOLAS, D. G. Context-based media adaptation in pervasive computing. In: *Proc. of Canadian Conference on Electrical and Computer Engineering*. [s.n.], 2001. Disponível em: <<http://citepseer.ist.psu.edu/446978.html>>.
- [Liu et al. 2001]LIU, G. C. et al. Standards for the electronic health recordemerging from health care tower of babel. In: *Proc. of AMIA Annual Symposium*. [S.l.: s.n.], 2001. p. 388–392.
- [Lorincz et al. 2004]LORINCZ, K. et al. Sensor networks for emergency response: Challenges and opportunities. In: *IEEE Pervasive Computing*. [S.l.: s.n.], 2004. v. 03, n. 4, p. 16–23.
- [Mahgoub e Ilyas 2004]MAHGOUB, I.; ILYAS, M. Mobile computing handbook. In: _____. [S.l.]: CRC Pr I Llc, 2004. cap. 15.
- [Meyer 1983]MEYER, P. *Probabilidade: Aplicações à Estatística*. [S.l.]: LTC Livros Técnicos e Científicos, 1983.
- [Neto 2006]NETO, F. A. *Uma Arquitetura de Suporte a Ciência de Contexto em TV Digital Móvel*. Dissertação (Mestrado) — UFPE, 2006.
- [Neves 1996]NEVES, J. L. Pesquisa qualitativa - características, usos e possibilidades. In: *Caderno de Pesquisas em Administração*. [S.l.: s.n.], 1996. v. 1, n. 3.
- [Novaes e Barbosa 2001]NOVAES, M. de A.; BARBOSA, A. K. P. Healthnet : um sistema integrado de telediagnóstico e segunda opinião médica. In: *III WORKSHOP RNP2 DURANTE O 190. SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES*. [S.l.: s.n.], 2001.

- [Novaes et al. 2004]NOVAES, M. de A. et al. Telehealth in northeast brazil: a pilot program for the public sector. In: *MEDINFO*. [S.l.: s.n.], 2004.
- [Pascoe 1998]PASCOE, J. Adding generic contextual capabilities to wearable computers. In: *2nd International Symposium on Wearable Computers*. [s.n.], 1998. p. 92–99. Disponível em: <citepseer.ist.psu.edu/pascoe98adding.html>.
- [Phan T.; Guy e Bagrodia 2001]PHAN T.; GUY, R. G.; BAGRODIA, R. A scalable, distributed middleware service architecture to support mobile internet applications. In: *Wireless Mobile Internet*. [s.n.], 2001. p. 27–33. Disponível em: <citepseer.ist.psu.edu/phan01scalable.html>.
- [PUCRJ 2005]PUCRJ. *MoCA Event-Based Communication Interface*. 2005. [Http://www.lac.inf.puc-rio.br/moca/event-service/index.html](http://www.lac.inf.puc-rio.br/moca/event-service/index.html).
- [Queiroz 1988]QUEIROZ, M. I. P. Experimentos com histórias de vida (itália-brasil). In: _____. [S.l.]: Vertice, 1988. cap. Relatos orais: do "indizível" ao "dizível", p. 68–80.
- [Rocha e Endler 2005]ROCHA, R. C. A. da; ENDLER, M. Evolutionary and efficient context management in heterogeneous environments. In: *MPAC '05: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*. New York, NY, USA: ACM Press, 2005. p. 1–7. ISBN 1-59593-268-2.
- [Rodriguez et al. 2004]RODRIGUEZ, M. D. et al. Location-aware access to hospital information and services. *IEEE Transactions on Information Technology in Biomedicine*, v. 8, n. 4, p. 448–455, 2004.
- [Roman 2002]ROMAN, M. Gaia: a middleware infrastructure to enable active spaces. In: *IEEE Pervasive Computing*. [S.l.: s.n.], 2002.
- [Rosenberg et al. 2002]ROSENBERG, J. et al. *SIP: Session Initiation Protocol*. [S.l.], 2002.
- [Ryan e Pascoe 1997]RYAN, D. M. N.; PASCOE, J. Enhanced reality fieldwork: the context-aware archaeologist assistant. In: GAFFNEY V., v. L. M. E. S. (Ed.). *Computer Applications and Quantitative Methods in Archaeology*. [S.l.: s.n.], 1997.
- [Sacramento et al. 2004]SACRAMENTO, V. et al. Moca: A middleware for developing collaborative applications for mobile users. *IEEE Distributed Systems Online*, 2004.

- [Saha e Mukherjee 2003]SAHA, D.; MUKHERJEE, A. Pervasive computing: A paradigm for the 21st century. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 36, n. 3, p. 25–31, 2003. ISSN 0018-9162.
- [Salber D. A.;Dey e Abowd 1998]SALBER D. A.;DEY, A.; ABOWD, G. D. *Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm*. [S.l.], 1998.
- [Satyanarayanan 1990]SATYANARAYANAN, M. Scalable, secure, and highly available distributed file access. *IEEE Computer*, v. 23, n. 5, May 1990.
- [Satyanarayanan 1996]SATYANARAYANAN, M. Mobile information access. *IEEE Personal Communications*, v. 3, n. 1, February 1996.
- [Schilit B.; Adams e Want 1994]SCHILIT B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: IEEE (Ed.). *IEEE Workshop on Mobile Computing Systems and Applications*. [S.l.: s.n.], 1994.
- [Schilit B.; Hilbert e Trevor 2002]SCHILIT B.; HILBERT, D.; TREVOR, J. Context-aware communication. *IEEE Wireless Communications*, p. 46–54, 2002.
- [Schilit 1995]SCHILIT, W. N. *A System Architecture for Context-Aware Mobile Computing*. Tese (Doutorado) — Graduate School of Arts and Sciences, Columbia University, 1995.
- [Schmidt 2000]SCHMIDT, D. Trends in distributed object computing. *Parallel and Distributed Computing Practices*, v. 3, n. 1, 2000.
- [Shacham et al. 2005]SHACHAM, R. et al. The virtual device: Expanding wireless communication service through device discovery and session mobility. In: *IEEE WiMob*. [S.l.: s.n.], 2005.
- [Sharmin e Ahmed 2006]SHARMIN, M.; AHMED, S. Marks (middleware adaptability for resource discovery, knowledge usability and self-healing) for mobile devices of pervasive computing environments. In: *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*. Washington, DC, USA: IEEE Computer Society, 2006. p. 306–313. ISBN 0-7695-2497-4.
- [Sousa e Garlan 2002]SOUSA, J. P.; GARLAN, D. Aura: an architectural framework for user mobility in ubiquitous computing environments. In: *In Proceeding of the 3rd Working IEEE/IFIP Conference on Software Architecture*. [S.l.: s.n.], 2002.

- [Stevenson 2001] STEVENSON, W. J. *Estatística: Aplicada à Administração*. [S.l.]: HARBRA, 2001. ISBN 8529400925.
- [Strang e Linnhoff-Popien 2004] STRANG, T.; LINNHOFF-POPIEN, C. *A Context Modeling Survey*. Sep 2004. Disponível em: <citepseer.ist.psu.edu/strang04context.html>.
- [Tanenbaum 2001] TANENBAUM, A. *Distributed Systems: Principles and Paradigms*. [S.l.]: Prentice Hall, 2001.
- [Tentori e Favela 2008] TENTORI, M.; FAVELA, J. Activity-aware computing for healthcare. *Pervasive Computing*, v. 7, n. 2, p. 51–57, April 2008.
- [Thakolsri et al. 2006] THAKOLSRI, S. et al. Accounting management for session mobility in an ubiquitous environment. In: *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*. New York, NY, USA: ACM Press, 2006. p. 311–316. ISBN 1-59593-306-9.
- [Trinta 2007] TRINTA, F. A. M. *Definindo e Provendo Serviços de Suporte à Jogos Multiusuário e Multiplataforma: Rumo à Pervasividade*. Tese (Doutorado) — Centro de Informática - UFPE, 2007.
- [Triola 1999] TRIOLA, M. *Introducao a Estatistica*. [S.l.]: LTC - Livros Tecnicos e Cientificos Editora, 1999.
- [Vieira et al. 2006] VIEIRA, V. et al. Tópicos em sistemas interativos e colaborativos. In: _____. [S.l.]: UFSCAR, 2006. cap. Uso e Representação de Contexto em Sistemas Computacionais, p. 127–166.
- [Vieira et al. 2007] VIEIRA, V. et al. Investigating the specifics of contextual elements management: The cementika approach. In: *CONTEXT 2007, LNAI*. [S.l.: s.n.], 2007. p. 493–506.
- [Voidsa 2002] VOIDA, e. a. S. Integrating virtual and physical context to support knowledge workers. *IEEE Pervasive Computing*, v. 1, n. 3, 2002.
- [Wegdam 2005] WEGDAM, M. Awareness: A project on context aware mobile networks and services medical systems international. In: *In the Proceedings of the 14th Mobile and Wireless Communications Summit 2005*. [S.l.: s.n.], 2005.
- [Weiser M.;Gold e Brown 1999] WEISER M.;GOLD, R.; BROWN, J. S. The origins of ubiquitous computing research at parc in the late 1980s. In: *IBM Syst. J.* [S.l.]: IBM Corp., 1999. v. 38, n. 4, p. 693–696.

- [Xiaosheng T.; Qinghua e Ping 2006]XIAOSHENG T.; QINGHUA, S.; PING, Z. A distributed context aware model for pervasive service environment. In: *The 1st International Symposium on Wireless Pervasive Computing - ISWPC*,. [S.l.: s.n.], 2006.
- [Yamin 2004]YAMIN, A. C. *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Junho 2004.

APÊNDICE A

HISTOGRAMAS

Este apêndice apresenta os histogramas obtidos através do software SPSS para os três experimentos realizados para cada uma das duas telas. São exibidos os histogramas gerados para 10 e 1000 sessões em cada experimento com cada tela.

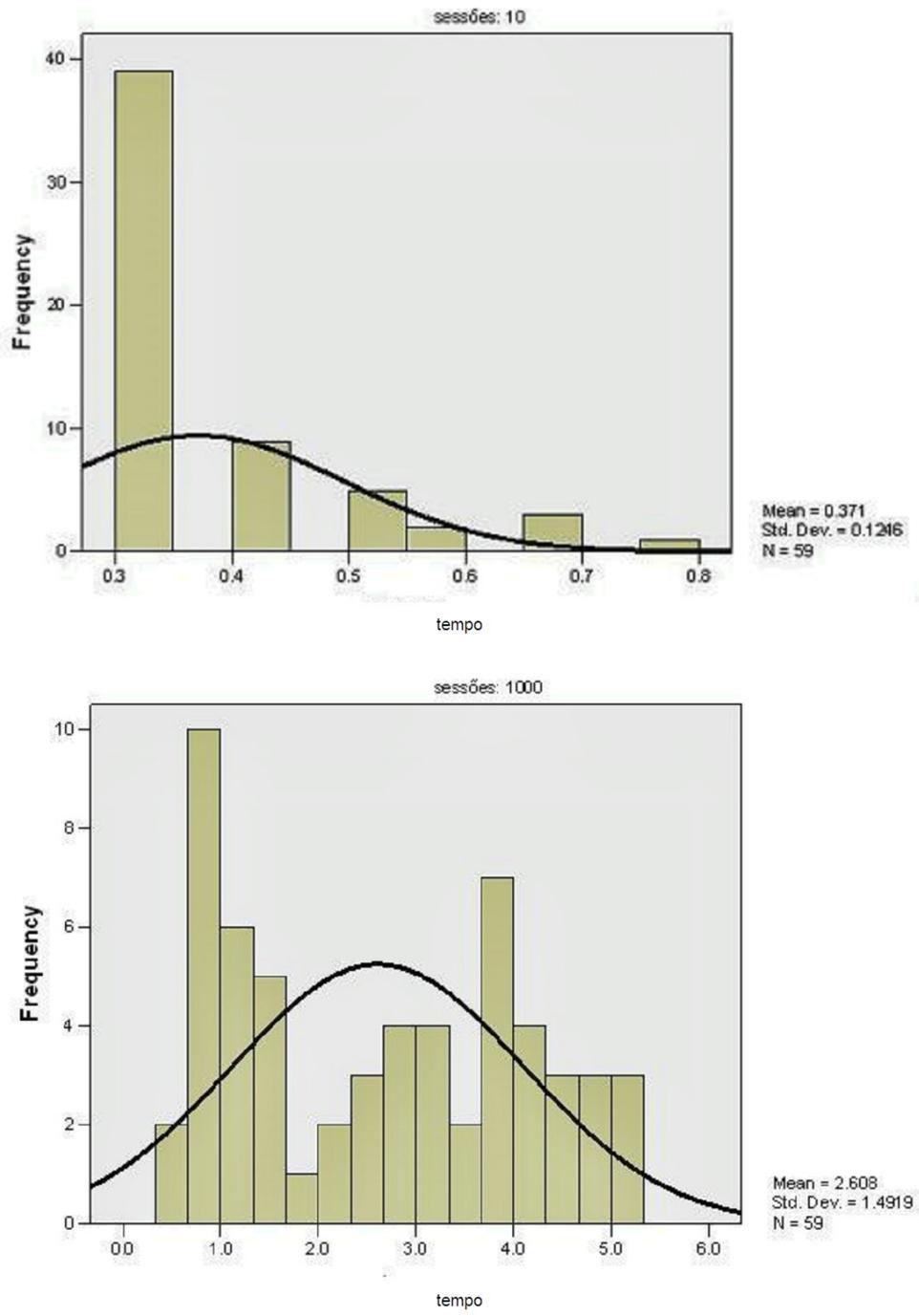


Figura A.1 Histograma do Tempo de Migração - Tela Consulta para Experimento 1

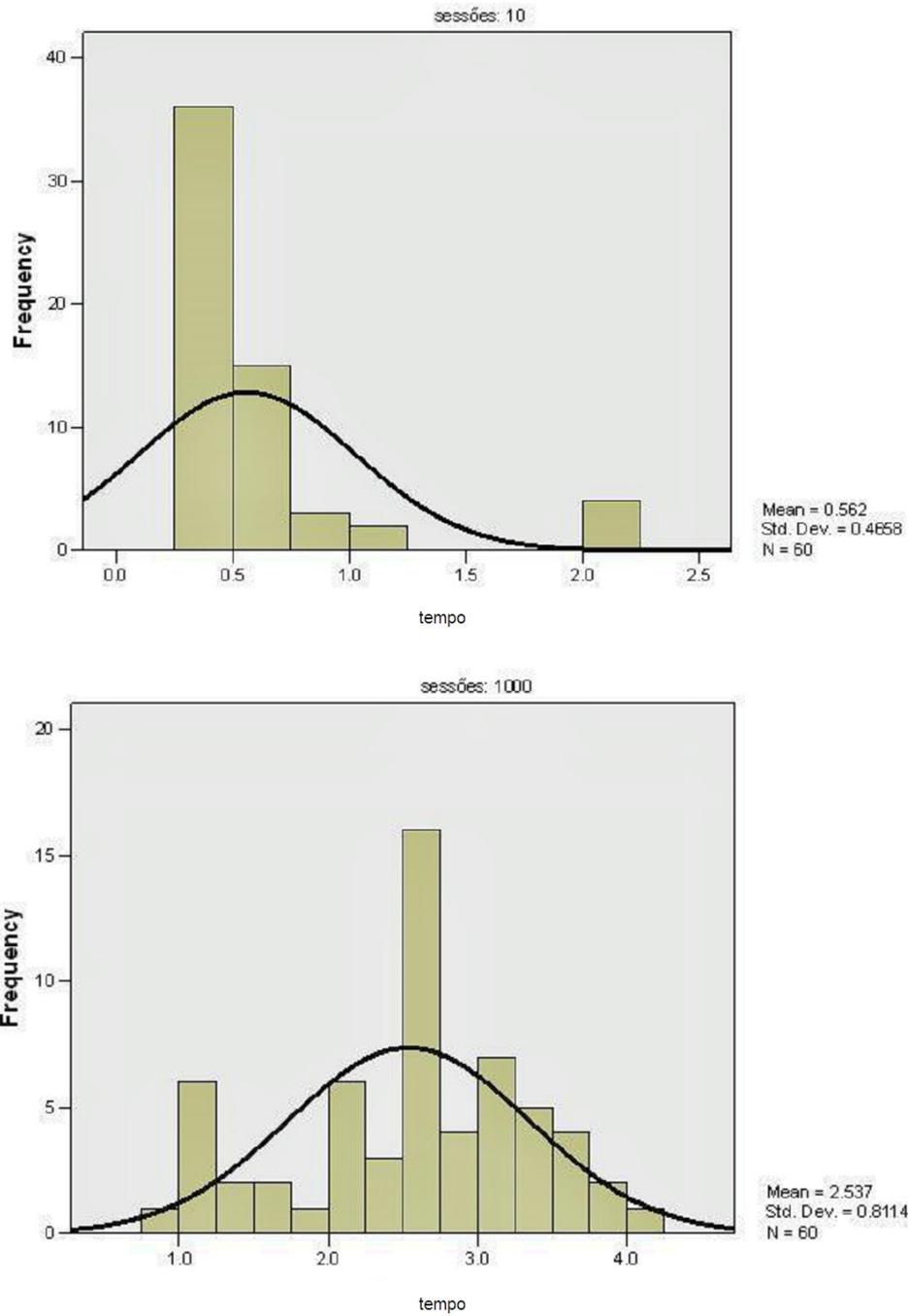


Figura A.2 Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 1

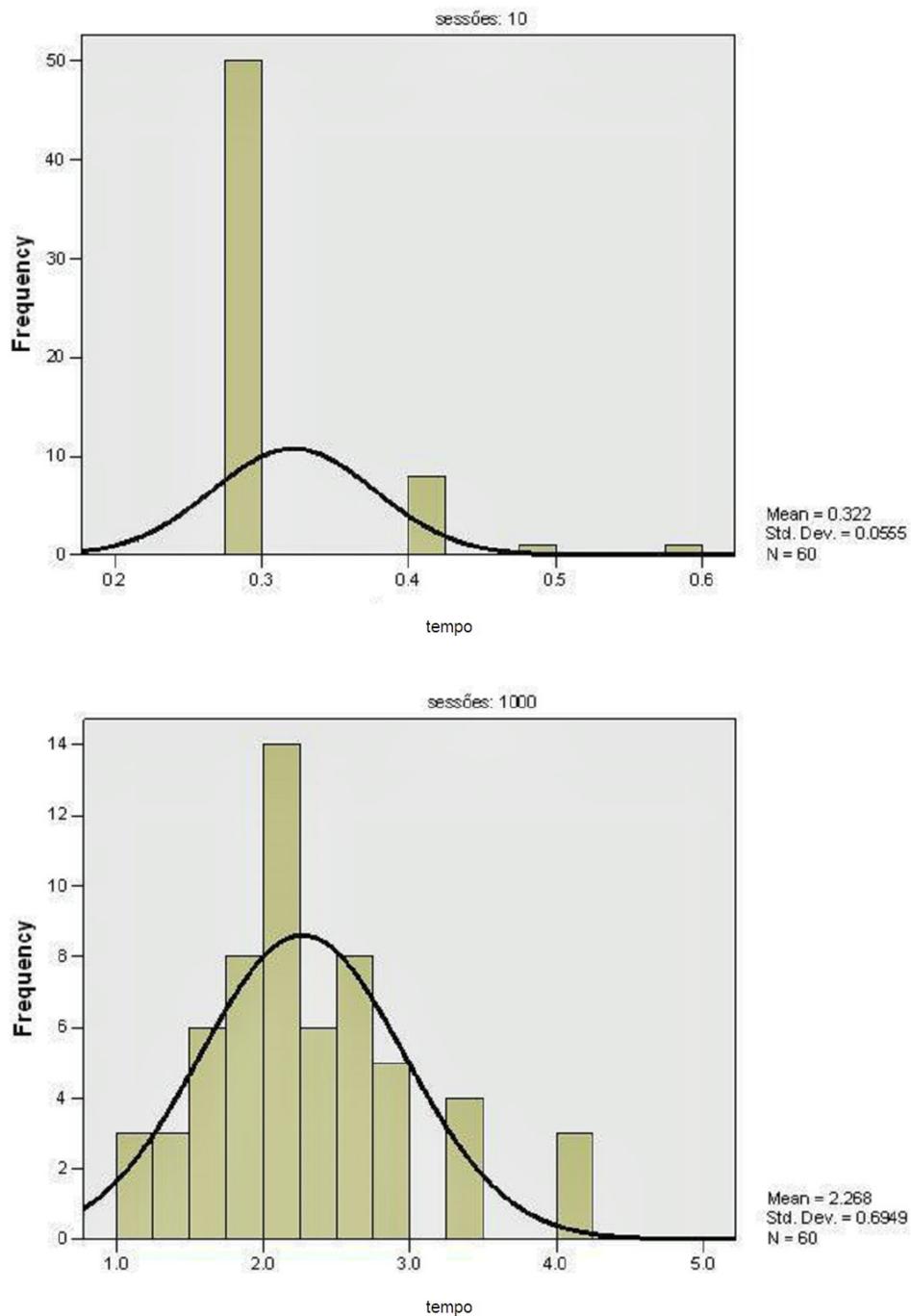


Figura A.3 Histograma do Tempo de Migração - Tela Consulta para Experimento 2

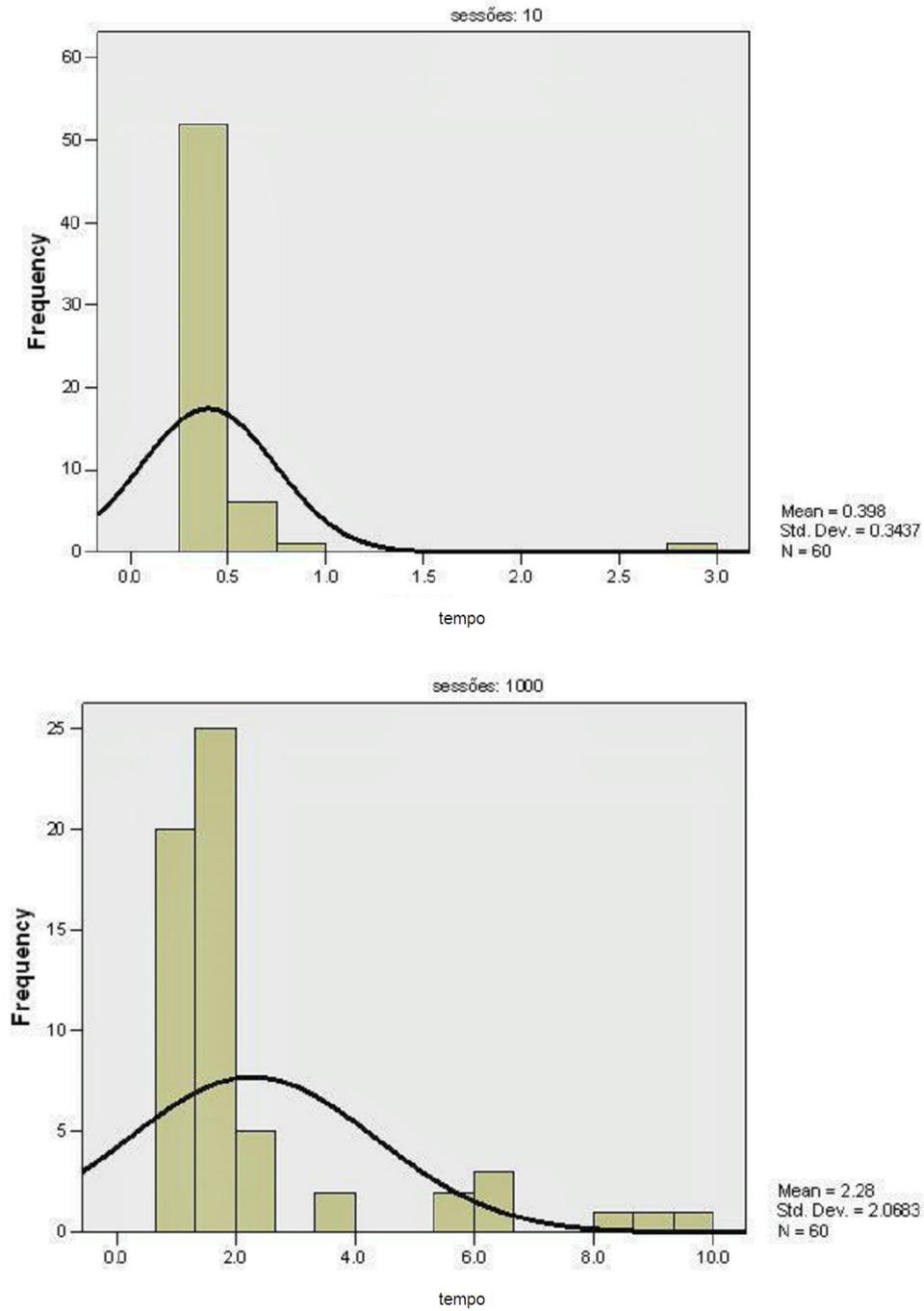


Figura A.4 Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 2

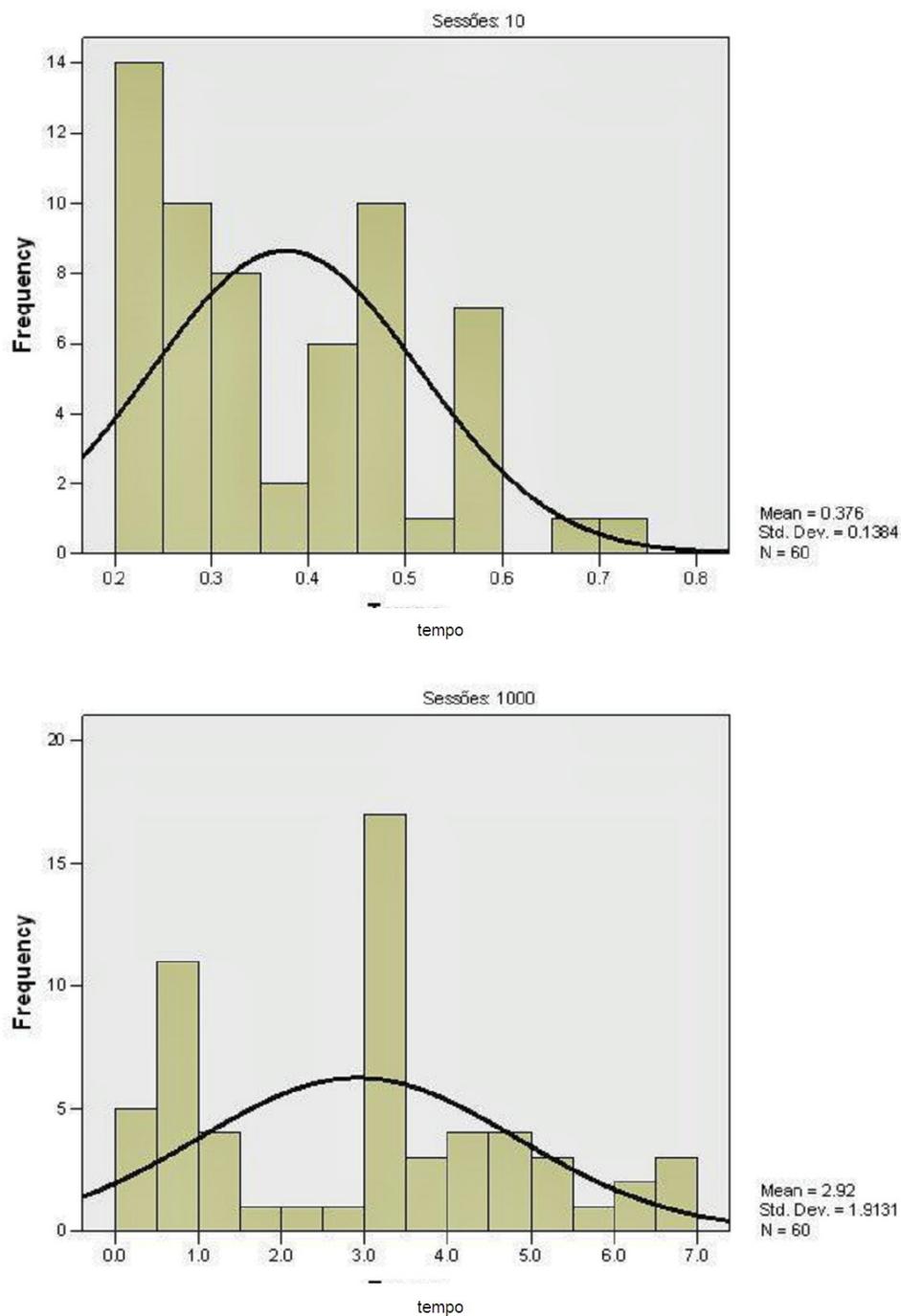


Figura A.5 Histograma do Tempo de Migração - Tela Consulta para Experimento 3

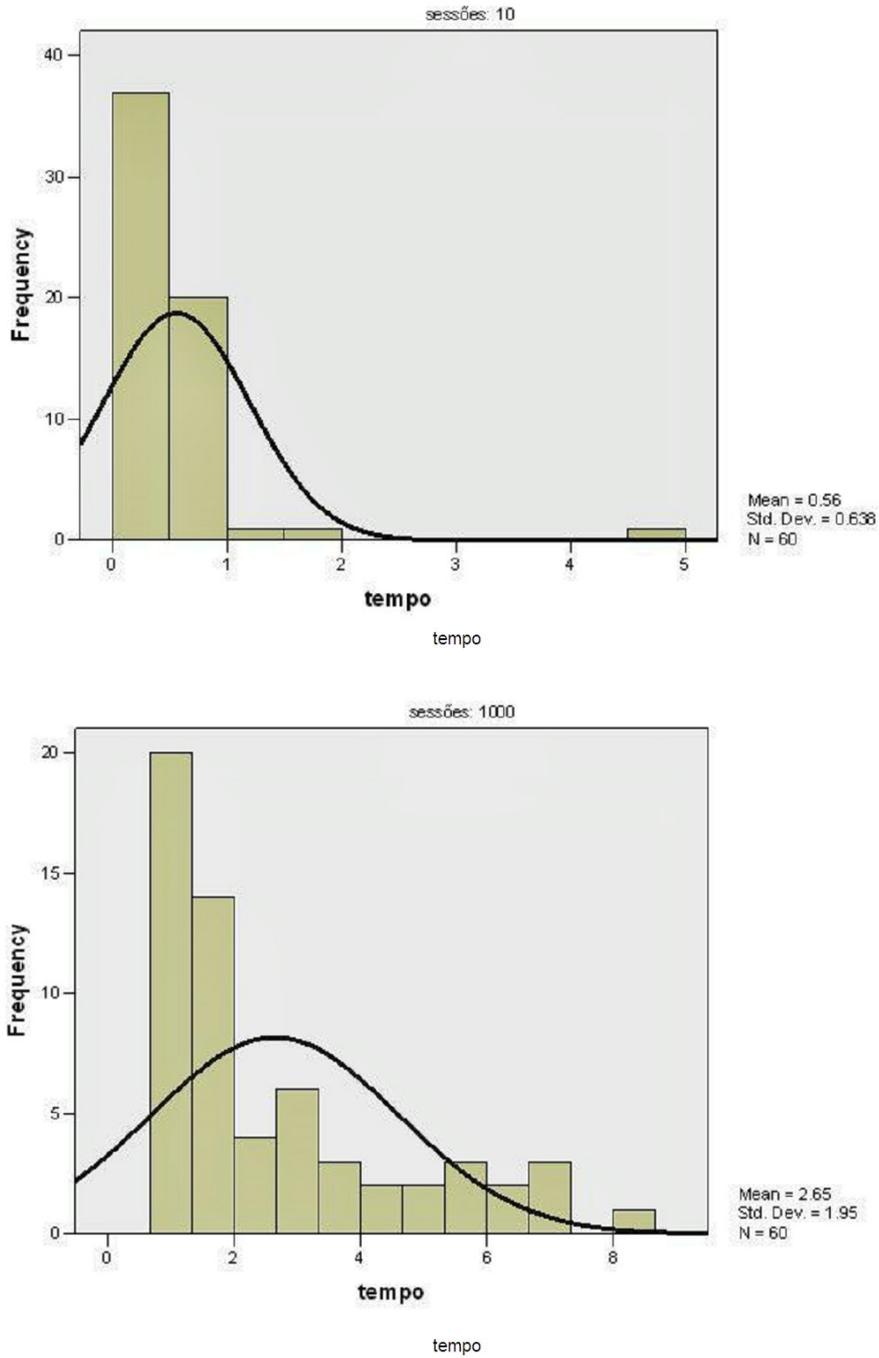


Figura A.6 Histograma do Tempo de Migração - Tela Responder Parecer para Experimento 3

APÊNDICE B

TERMO DE COMPROMISSO

Você está sendo convidado a participar de um experimento para identificar as impressões do usuário em relação à possibilidade de aumento na produtividade do trabalho médico em sua rotina diária. Este experimento faz parte de uma tese de doutorado desenvolvida no Centro de Informática da Universidade Federal de Pernambuco, que propõe dar suporte computacional aos usuários que tem como característica das suas atividades diárias, a mobilidade e fragmentação do trabalho.

Garantimos a todos os participantes o seu anonimato (ele não será identificado em nenhum resultado divulgado da pesquisa) e a confidencialidade de dados (nenhuma informação pessoal ou capaz de identificar o participante será divulgada pelos pesquisadores). Além disto, a participação no teste é rigorosamente voluntária, não estando o participante obrigado sequer a concluir as atividades propostas - pode interromper sua participação a qualquer momento, sem estar de forma alguma obrigado a justificar ou explicar a interrupção. Caso você consinta em participar deste teste, pedimos que assine as duas vias deste documento e guarde uma com você.

Somos gratos a sua contribuição para a nossa pesquisa, e estamos a seu dispor para quaisquer outros esclarecimentos.

Recife,(data) de agosto de 2008

Assinatura do Participante

Juliana R. Basto Diniz (Aluna de Doutorado)

Carlos André G. Ferraz (Orientador)

QUESTIONÁRIO DE AVALIAÇÃO

C.1 Identificação do Usuário

1. Qual o seu nível de conhecimento em Informática?

ALTO MÉDIO BAIXO

2. Já fez uso de sistemas de prontuários eletrônico de pacientes?

SIM NÃO

3. Já fez uso de aplicações em dispositivos móveis? Quais aplicações? Quais dispositivos?

SIM NÃO

C.2 Uso de PEPs

1. Você acha que um sistema de informação que dê acesso a prontuários eletrônicos distribuídos nos diversos hospitais credenciados a uma rede, facilitaria a sua vida? Se Sim, em que aspectos?

SIM NÃO

2. Na sua opinião, quais as atividades médicas mais comuns que você realiza e que necessitam da consulta a PEPs?

3. Você já vivenciou alguma situação onde precisou fazer acesso a um sistema de prontuários eletrônicos usando dispositivos móveis? Em que situação?

SIM NÃO

4. Dentre as funcionalidades de um sistema de prontuário eletrônico apresentadas abaixo quais deveriam ser disponibilizadas numa interface de um dispositivo móvel? Coloque E para as que julga essenciais, D para as desejáveis e I para irrelevantes.

- Alterar seus dados pessoais
- Incluir um novo paciente
- Consultar os dados de um paciente cadastrado
- Acrescentar informações aos dados demográficos de um paciente
- Acrescentar informações dos antecedentes de um paciente
- Acrescentar um novo caso clínico a um paciente
- Consultar um caso clínico específico
- Acrescentar informações a um caso clínico já existente
- Solicitar uma segunda opinião (parecer) de um caso clínico
- Consultar pareceres de um caso clínico
- Responder os pareceres (segundas opiniões) que lhes forem solicitados
- Visualizar o cadastro de médicos do sistema
- Visualizar o cadastro de hospitais e unidades de saúde cadastradas

Cite outras funcionalidades que julgue essenciais:

Cite outras funcionalidades que julgue desejáveis:

C.3 Sobre os Serviços

1. Para você seria útil que o sistema permitisse migrar uma sessão em execução para um outro dispositivo? Caso tenha respondido SIM, indique em que situações (Figura C.1) você classifica como útil o processo de migração.

SIM NÃO

2. Com relação a velocidade, como você classifica o processo do resgate de uma sessão em migração?

RÁPIDO LENTO MUITO LENTO

3. O conteúdo mostrado nos campos apresentados é considerado suficiente para a sua análise de um caso clínico? Por que?

SIM NÃO

Computador de mesa/notebook → Celular

- Precisa sair do consultório ou residência e gostaria de terminar o trabalho em seu celular.
- Continuara dentro do hospital, mas recebeu um chamado de urgência para atender um paciente presencial e pode concluir o trabalho do seu celular enquanto desloca-se ao encontro do paciente.

Celular → Computador de mesa/notebook

- Precisa terminar o trabalho e bateria do seu celular está acabando.
- Deseja continuar trabalhando em um dispositivo que permita melhor visualização do prontuário eletrônico de pacientes.
- Deseja digitar muitas informações e prefere fazê-lo em um notebook ou computador de mesa.

Computador de mesa/notebook → Computador de mesa/notebook

- Precisa sair do consultório/hospital e gostaria de continuar o trabalho em casa, algumas horas depois.
- Precisa sair do consultório, hospital ou residência e gostaria de terminar o trabalho em seu notebook no restaurante/refeitório.

Celular → Celular

- A bateria do seu celular está acabando e precisa terminar o trabalho usando outro celular.

Figura C.1 Situações onde a Migração poderia ser Útil

4. Você acredita que este ambiente pode aumentar a produtividade do seu trabalho?

SIM NÃO